



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH INFORMAČNÍHO SYSTÉMU

INFORMATION SYSTEM DESIGN

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Zdeněk Bartušek

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Miloš Koch, CSc.

BRNO 2020

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **Zdeněk Bartušek**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Manažerská informatika
Vedoucí práce: **doc. Ing. Miloš Koch, CSc.**
Akademický rok: 2019/20

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

Návrh informačního systému

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza problému
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Analyzovat stávající stav informačního systému vybrané organizace a jeho efektivnosti, posoudit tento stav a na základě firemní strategie připravit návrh řešení nového informačního systému.

Základní literární prameny:

BASL, Josef a Roman BLAŽÍČEK. Podnikové informační systémy: podnik v informační společnosti. 3. aktualiz. a dopl. vyd. Praha: Grada, 2012. 323 s. ISBN 978-80-247-4307-3.

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 2. přeprac. a aktualiz. vyd. Praha: Grada, 2009. 496 s. ISBN 978-80-247-2615-1.

MOLNÁR, Zdeněk. Efektivnost informačních systémů. 2. rozš. vyd. Praha: Ikar, 2000. 178 s. ISBN 80-247-0087-5.

SCHWALBE, Kathy. Řízení projektů v IT. Brno: Computer Press, 2007. 720 s. ISBN 978-80-251-1-26-8.

SODOMKA, Petr a Hana KLČOVÁ. Informační systémy v podnikové praxi. 2. aktualiz. a rozš. vyd. Brno: Computer Press, 2010. 501 s. ISBN 978-80-251-2878-7.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2019/20

V Brně dne 29.2.2020

L. S.

doc. RNDr. Bedřich Půža, CSc.
ředitel

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

Abstrakt

Náplní této bakalářské práce je návrh části informačního systému pro FN Brno. V první části práce jsou definovány základní teoretické pojmy, jejichž znalost je využita při tvorbě návrhu. Druhá část obsahuje analýzu současného stavu ve FN Brno a jejich požadavků na samotný systém. Ve třetí části dochází k tvorbě a popsání návrhu dle informací, které byly získány z analýz a požadavků zadavatele.

Klíčová slova

informace, databáze, data, analýza, informační systém, SWOT analýza, procesy

Abstract

The content of this thesis is to design parts of the information system for the FN Brno. The first part defines the basic theoretical concepts that can be used in designing. The second part contains the advantage of the current situation in the FN Brno and their requirements for the system itself. In the third part of the slide, a proposal is created and described that contains the information you have available and that you enter.

Key words

information, database, data, analysis, information system, SWOT analysis, processes

Bibliografická citace

BARTUŠEK, Zdeněk. *Návrh informačního systému* [online]. Brno, 2020 [cit. 2020-05-13]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/124787>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Miloš Koch.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 13. května 2020

.....

podpis studenta

Poděkování

Tímto bych rád poděkoval především svému vedoucímu práce doc. Ing. Miloši Kochovi, CSc., jehož rady a odborné informace mi velmi pomohly při tvorbě této práce. Zároveň bych chtěl poděkovat všem zaměstnancům firmy CSystem, kteří mi poskytli potřebné informace z firemního a nemocničního prostředí a také přátelům a rodině, kteří mě při tvorbě práce podporovali.

OBSAH

| | |
|---|-----------|
| OBSAH | 8 |
| ÚVOD | 10 |
| 1 TEORETICKÁ VÝCHODISKA PRÁCE..... | 12 |
| 1.1 Data | 12 |
| 1.2 Informace | 13 |
| 1.3 Informační zdroje | 13 |
| 1.4 Informační systém..... | 15 |
| 1.5 Nemocniční informační systém..... | 16 |
| 1.5.1 Architektura NIS | 17 |
| 1.5.2 Integrace aplikací NIS | 18 |
| 1.6 Datové a komunikační standardy | 19 |
| 1.6.1 Český datový standard DS | 20 |
| 1.6.2 Standard HL7 | 20 |
| 1.7 Datový sklad..... | 21 |
| 1.7.1 Systém datového skladu..... | 21 |
| 1.8 Business intelligence | 23 |
| 1.9 Nástroje InterSystems | 23 |
| 1.9.1 Caché | 23 |
| 1.9.2 Ensemble..... | 24 |
| 1.9.3 DeepSee | 25 |
| 2 ANALÝZA SOUČASNÉHO STAVU | 27 |
| 2.1 SLEPT analýza..... | 27 |
| 2.1.1 Sociální faktory..... | 27 |
| 2.1.2 Legislativní faktory..... | 28 |
| 2.1.3 Ekonomické faktory..... | 29 |
| 2.1.4 Politické faktory..... | 29 |
| 2.1.5 Technologické faktory | 29 |
| 2.2 Analýza 7S | 30 |
| 2.2.1 Strategie | 30 |
| 2.2.2 Struktura..... | 30 |

| | | |
|----------|---|-----------|
| 2.2.3 | Systémy..... | 31 |
| 2.2.4 | Styl..... | 33 |
| 2.2.5 | Sdílené hodnoty | 33 |
| 2.2.6 | Spolupracovníci | 34 |
| 2.2.7 | Schopnosti..... | 34 |
| 2.3 | SWOT analýza | 34 |
| 2.4 | Analýza požadavků zadavatele | 36 |
| 3 | VLASTNÍ NÁVRHY ŘEŠENÍ..... | 39 |
| 3.1 | Architektura žádankového portálu | 39 |
| 3.2 | Business vrstva..... | 40 |
| 3.2.1 | Případy užití..... | 41 |
| 3.3 | Aplikační vrstva | 60 |
| 3.3.1 | Realizace případů užití..... | 60 |
| 3.4 | Ekonomické zhodnocení projektu..... | 74 |
| 3.4.1 | Náklady řešení | 75 |
| 3.4.2 | Přínosy spojené s nasazením systému | 76 |
| | ZÁVĚR | 77 |
| 4 | SEZNAM POUŽITÝCH ZDROJŮ | 78 |
| 5 | SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ | 79 |
| 6 | SEZNAM POUŽITÝCH OBRÁZKŮ | 80 |
| 7 | SEZNAM POUŽITÝCH TABULEK | 81 |

ÚVOD

Dnešní doba je neodmyslitelně spjata s elektronizací ať už dat či různých úkonů, které se tím stávají rychleji proveditelné a my jich můžeme zvládat větší množství v kratším časovém horizontu než tomu bylo doposud. To vše přispívá k neustálému zrychlování tempa našich životů. O tom, že jsme téměř všichni nějakým způsobem závislí na určitém druhu elektroniky, na službách či informacích, které nám poskytuje, není nutno dlouze debatovat, mnohdy stačí chvilkový výpadek elektřiny a rázem se ocitáme mimo naši komfortní zónu. Naše životy jsou tedy spřaženy s využíváním těchto technologií a při správném zacházení nám ho mohou velice usnadňovat.

Jenom v málo odvětvích bychom pak našli tak těsnou symbiózu mezi elektronickým systémem a informacemi, které poskytuje, a člověkem, jako ve zdravotnictví. Tato závislost přitom může být mnohdy doslova životně důležitá a není zde prostor na chyby.

Chtěl bych se zmínit především o samotném informačním systému nemocnice, který považuji za jednu z klíčových věcí pro její fungování a celkový chod. Ještě donedávna, a v mnoha zařízeních tomu tak ještě stále je, se informace v těchto systémech udržovaly v tištěné podobě. To přinášelo hned několik nevýhod oproti elektronickému řešení, a to především vyšší časovou náročnost při vyhledávání informací, větší požadavky na prostor pro uložení dat, mnohem nižší integraci dat atp. Celý informační systém má mnoho kritérií, která musí splňovat a musí být dostatečně robustní, aby mohl pracovat s tak velkými objemy informací, které se v tomto prostředí shromažďují.

V mojí bakalářské práci se tedy budu věnovat právě návrhu jedné z částí tohoto rozsáhlého nemocničního informačního systému.

CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ

Hlavním cílem této práce bude navržení části informačního systému - konkrétně žádankového portálu - pro Fakultní nemocnici Brno. Díky tomuto portálu budou moci oprávnění uživatelé manipulovat (vytvářet, editovat, zobrazovat, tisknout, odesílat atd..) s různými druhy žádanek, celý systém by měl být navržen a zpracován tak, aby zvládal bezpečně pracovat s velmi citlivými daty, která se v něm nacházejí a zároveň, aby i práce s tímto systémem byla pohodlná pro běžné uživatele a nepůsobila jim víc obtíží než současně používaná aplikace.

Moji práci jsem rozčlenil na tři hlavní části, které bych chtěl v následujících odstavcích popsat.

V první části se věnuji popsání základních pojmů, spolu s teoretickým minimem, které je nezbytné pro základní orientaci v tomto tématu. Vše poslouží pro lepší uvedení do problematiky návrhu informačního systému a načerpané informace budou využity i v dalších částech práce.

Druhá část se zaměřuje na provádění analýz pro zjištění všech vlivů, které mohou mít vliv na samotný informační systém, aby bylo zahrnuto jejich působení při jeho návrhu. Mezi použitými analýzami jsou SWOT, 7S a SLEPT analýza. Zároveň v této části uvážím a zapracuji požadavky zadavatele.

V poslední, třetí části, se budu věnovat návrhu žádankového portálu dle požadavků zadavatele, tak, aby splňoval všechny požadavky na funkčnost. Budu vycházet především z vytvořeného diagramu pro návrh této aplikace. Na úplný závěr pak přidám celkové ekonomické zhodnocení tohoto projektu a přínosy tohoto projektu.

1 TEORETICKÁ VÝCHODISKA PRÁCE

V této části mé práce se budu zabývat teoretickým pozadím pro pochopení mnou zvoleného tématu práce. Budou zde sepsány a vysvětleny základní pojmy z oblasti teorie potřebné pro návrh informačního systému. Na základě znalostí těchto informací bude snazší dané problematice porozumět, případně mohou pomoci lépe zhodnotit mnou předložený návrh.

1.1 Data

Pro IT a pro případné počítačové zpracování si většinou pod tímto pojmem můžeme představit obrázky, zvuk, text, čísla, nebo jiné smyslové vjemy, které jsou vhodné k tomuto účelu. (1)

Data dělíme do následujících skupin:

- **Strukturovaná data**

Jedná se o data, jejichž struktura je jasně definována, například při ukládání pomocí relačních databázových systémů, kdy se využívá specifické hierarchie elementů. Právě proto jsme schopni při potřebě získání dat vybrat pouze ta data, která odpovídají našim požadavkům.

- **Nestrukturovaná data**

Tento typ dat je reprezentován jako „tok bitů“, které nemají žádné další rozlišení. Jako příklad těchto dat může být chápán obrázek, případně audio nebo video nahrávka. (1)

Mnou zvolená definice pojmu data pochází od Viléma Sklenáka, který tento pojem vnímal takto: „Data slouží pro reprezentaci faktů, atributů, odrazu dějů a věcí.“ (1, s. 2)

Data sama o sobě, bez dalšího kontextu, by nám nemusela dávat smysl, avšak při zasazení do správného kontextu, případně nabytí více poznatků o určitých datech, jim dává alespoň nějakou srozumitelnou výpovědní hodnotu, informaci. Data jsou tedy základem pro informace. (1)

1.2 Informace

„Informace jsou data v kontextu, jsou to data použitelná a srozumitelná.“ (1, s. 3)

Zmíněná definice je pouze jednou z mnoha, jelikož záleží, ve kterém oboru se nacházíme, pro každý může tento pojem znamenat něco odlišného. Obecně lze ale říci, že pokud k nějakým datům přidáme sémantiku, vznikne nám určitý poznatek, jenž není tak abstraktní a je lépe spojitelný s realitou. Na základě těchto poznatků pak máme informace. Zásadním rozdílem mezi informacemi a poznatky je, že informace je časově závislá, na rozdíl od poznatku, jenž je v čase trvalý. Vztah těchto dvou pojmů lze také popsat tak, že informace jsou jakékoliv používané poznatky. (1)

Každý využívá jiný druh informací, pokud vztáhneme použití informací na jednotlivé lidi, tak by se dalo říci, že každý využívá svoje specifické druhy. To, které si vybírá, může být ovlivněno mnoha faktory - subjektivními znalostmi, které se zakládají na zkušenostech, názorech, hodnotách atd. S tím může souviset i **hodnota informace**, která je součástí přeměny dat na informace a je opět subjektivní pro každého člověka. Tato hodnota však nemá s cenou dat mnoho společného, data se dají nakupovat za horentní sumy, avšak až při jejich použití zjišťujeme skutečnou hodnotu informace, kterou tato data nesou. Pokud se například ukáže, že nakoupená data jsou zastaralá nebo poškozená, nepřinesou nám žádnou informaci, tudíž ani hodnotu. (1)

1.3 Informační zdroje

I když je přesná definice pojmu poměrně složitá, lze ji uvést podle Sklenáka takto *„Informačním zdrojem budeme rozumět systém, který je reálným, nebo potencionálním nositelem, zprostředkovatelem, nebo šířitelem informací.“* (1, s. 22) Jinými slovy to znamená jakýkoliv zdroj dat, který obsahuje a poskytne nám námi požadované informace, jako například televize, rozhlas, databázová centra nebo knihovny, jenž patří k velmi důležitým informačním zdrojům. (1)

Informační zdroje můžeme rozdělovat podle různých specifik, buď podle typu informací, které nám daný zdroj poskytne, což nám umožní dělit zdroje poskytující ekonomické, právníkové či politické informace a mnohé další, případně se vyjmenované zdroje dají dělit

na další, jemnější prvky. Další dělení může být podle toho, v jakém formátu nám zdroje dané informace poskytují, z toho vyplývá dělení na zdroje textových, obrazových, zvukových nebo multimediálních informací. Jako další dělení se pak nabízí dostupnost zdroje podle toho, zda-li jsou informace veřejně dostupné nebo se za ně musí zaplatit, což odpovídá komerčnímu zdroji. Jedná se o velmi hrubé rozdělení na základní typy zdrojů, jistě by se dala uvádět další a další, avšak pro nastínění významu je tento výčet dostačující. (1)

Při výběru informačního zdroje, který budeme používat k získávání pro nás potřebných informací, je důležité zvážit kvalitu poskytovaných informací a cenu, kterou za ně zaplatíme, teprve poté je dobré rozhodnout se pro určitý zdroj. Při zhodnocování kvality informačního zdroje je dobré zabývat se těmito vlastnostmi:

- **Typ informací** – značí, o jaké informace se jedná z hlediska typu, můžeme mít informaci bibliografickou, případně celý text dokumentu, případně faktografickou, obrazovou, zvukovou, multimediální atd. Výběr volíme podle naší informační potřeby.
- **Rozsah** – mluvíme-li o rozsahu zdroje, znamená to, kolik záznamů nebo jiných jednotek je v něm obsaženo. Obvykle je tomu i přímo úměrná cena informací.
- **Úplnost** – pokud mluvíme o úplnosti zdroje, tak se zaměřujeme na to, jak velké množství všech informací, kterými se zdroj zabývá, je ve zdroji uloženo. Tento faktor je důležitý například při patentové nebo právní informaci.
- **Retrospektiva zdroje** – u zdroje nám říká, jak daleko do minulosti můžeme datovat uchovávané informace v něm obsažené. Využití nachází např. u časových řad.
- **Perioda aktualizace** – zachycuje, s jakou frekvencí jsou do zdroje přidávány nové, aktuální informace.
- **Producent** – je jedním z určujících faktorů důvěryhodnosti zdroje, má svůj význam např. u databází. Opět se „věhlas“ producenta odráží přímo úměrně na ceně informací.
- **Dostupnost** – určuje, jestli je daný zdroj volně dostupný či zda-li je to komerční zdroj. S tímto také souvisí, jestli průzkum zdroje provádí sám zájemce o

informace, případně je-li průzkum zdroje prováděn provozovatelem, na čemž závisí i doba, za kterou se k nám informace dostanou.

- **Cena** – ta může být určována podle různých kritérií. Lze říci, že čím kvalitnější informace (renomovaný producent, velký rozsah a úplnost dat atd.), tím se cena pohybuje ve vyšších částkách. Faktory ovlivňující cenu informace byly uvedeny v předcházejících odstavcích. Cena může být určena paušální platbou, případně u dialogových databází může být účtováno za každý poskytnutý záznam, nebo se počítá celkový čas strávený v databázi, to už záleží na nastavení podmínek u každého poskytovatele zvlášť. (1)

1.4 Informační systém

V informačním systému hrají primární roli informace, jeho jednotlivé prvky jsou pak obvykle shodné s komponentami byznys systému. Informační systém lze též brát jako nedílnou součást byznys systému, avšak účel obou systémů je různý. (2)

Pro informační systém platí, že by měl doručovat správné informace, na určené místo v určený čas. Pod místem si můžeme představit konkrétního člověka, který ony informace zpracovává a je také zároveň součástí byznys systému, jinými slovy hovoříme o uživateli IS. To, jak dobře pracuje náš IS a jestli je vhodně využíván, zjistíme podle schopnosti správně podporovat náš byznys pro dosahování jeho cílů, kterými může být především dosahování zisků atp. (2)

Aby informační systém plnil dobře svoji úlohu, je třeba ho vybavit správnými informačními a komunikačními technologiemi (ICT). Systém s těmito komponentami pak obvykle nese označení IS/ICT. V byznys systému jsou ICT využívány taktéž, ale nenabývají takového významu, jako v informačním systému ve vztahu k informacím. (2)

„Informační a komunikační technologie (ICT) jsou hardwarové a softwarové prostředky pro sběr, přenos, ukládání, zpracování a distribuci a pro vzájemnou komunikaci lidí a technologických komponent IS.“ (3, s. 446)

Již jsem uvedl, že rozsah obou zmiňovaných systémů je téměř totožný, v poslední době se však vyplatí, aby samotný IS měl určitý „přesah“ a to ve smyslu rozšíření a zahrnutí do informačního systému i subjekty z okolí našeho byznysu, např. zákazníky. Ti totiž obvykle zpracovávají podobné, ne-li stejné informace jako my a zároveň mohou vytvářet důležité informace pro nás, pokud bychom tedy měli IS vztažen například pouze na náš podnik, mohli bychom přicházet o podstatnou část informací z okolí, případně by přístup k těmto informacím mohl být mnohem složitější. (2)

Zároveň však není nezvyklé, že se vytvoří IS pouze pro určitou část podniku, tyto systémy se pak mezi sebou mohou propojit, případně se nad nimi vytvoří další informační systém, který bude spojovat všechny části. Tyto varianty přinášejí ulehčení samotného návrhu, náhledu na informace, které z nich plynou, a jejich pochopení. (2)

„Na informační systém organizace se také můžeme dívat jako na systém informačních a komunikačních technologií, dat a lidí, jehož cílem je efektivní podpora informačních, rozhodovacích a řídicích procesů na všech úrovních řízení organizace.“ (3, s. 446)

1.5 Nemocniční informační systém

J.H. van Bommel a M.A. Musen uvádějí ve své knize jednu z možných definic takového nemocničního informačního systému (NIS). (5)

„Cílem NIS je využití počítačů a komunikačních prostředků ke sběru, uchovávání, zpracování, prezentaci a přenosu patientských a administrativních informací souvisejících se všemi činnostmi nemocnice a uspokojení funkčních potřeb autorizovaných uživatelů.“ (5, s. 107)

Vzhledem k uvedené definici je patrné, jak sofistikovaný takový systém musí být a jaké množství aspektů jej může ovlivňovat. Vývoj a používání informačních systémů začalo na přelomu šedesátých a sedmdesátých let minulého století, od té doby se tyto systémy velmi dynamicky změnily a stále rostou závratným tempem, spolu se zdravotnictvím jako celkem. V celém NIS je nejpodstatnější ta část, která zpracovává data o diagnostických a terapeutických procesech každého konkrétního pacienta. O této části pak lze mluvit jako o patientsky orientované. Jak již bylo zmíněno v definici, pouze tato část by však nestačila v komplexním NIS, kde jsou zpracovávána data z mnoha dalších oblastí, jako

například zásobování léky, mzdy, personalistika atd. Vše navíc musí pružně a efektivně fungovat a vyhovovat požadavkům uživatelů a samotné nemocnici s jejími procesy.

Vzhledem k uvedeným jednotlivým částem, které jsou součástí celého nemocničního prostředí, je třeba vše rozdělit do jednotlivých subsystémů či modulů, jenž je mezi sebou potřebné datově a funkčně provázat, čímž nám vzniká komplexní NIS. Varianty, kterými jsou jednotlivé aplikace potřebné pro provoz a klíčové procesy realizovány, závisí na zvolené architektuře NIS a především na tzv. referenčním modelu, který vymezuje data a vlastnosti aplikací s jejich vazbami. (5)

1.5.1 Architektura NIS

Funkční NIS je možné vytvořit na více variantách upořádání IT prostředků. To, jakou architekturu si vybereme, nemá vliv na funkční a užité vlastnosti aplikací nacházejících se v systému, ale výrazně to ovlivňuje procesy vývoje, nasazení a správy NIS. Hlavní rozdělení na jednotlivé architektury jsou: Monolitní, Evoluční a Kompozitní. (5)

Monolitní – tento typ architektury v sobě většinou skrývá velmi robustní řešení, v němž je celý NIS realizován jako jeden systém jedním výkonným serverem. Velkou výhodou tohoto typu architektury je její jednoduchá správa a údržba, problémová je však jeho implementace a samotný vývoj. Monolitní architektury se vytvářely především v minulosti, kvůli absenci výpočetního výkonu tehdejších strojů. Některé NIS jsou však na této architektuře v provozu dodnes. (5)

Evoluční – stejně jako předchozí architektura uchovává nejvytíženější část systému, obvykle klinický IS – CIS s nejvyšším počtem uživatelů jako centrální systém, který se nachází na hlavním serveru. Okolo něj jsou pak vytvořeny ostatní, problémověji orientované aplikace s menším počtem uživatelů, jako laboratoře – LIS, lékárna – LS, ekonomické, finanční a správní provozy – EIS atd., které jsou poměrně soběstačné IS s propojením na hlavní aplikaci ať už komunikačním modulem nebo interfacem. V současné době se jedná asi o nejrozšířenější typ architektury v NIS, i když se u některých implementacích může lišit počet modulů v centrálním systému. Výhodou této architektury pak je možnost vytvářet jednotlivé subsystémy různými dodavateli

s optimálním nastavením SW pro konkrétní procesy, kterou daná část systému vykonává. Pokud máme jednotlivé části vytvořeny třeba na rozdílných SW platformách od různých firem hovoříme pak o heterogenním systému. To se však může stát i nevýhodou a to především v oblasti komunikace jednotlivých částí mezi sebou a případně s centrálním serverem, jelikož různorodost nutí vytvářet složité komunikační moduly. Stejně tak implementace a správa nejsou tak snadné jako u Monolitní architektury. (5)

Kompozitní – je posledním typem architektury, v níž je každý funkční modul vybudován jako nezávislý subsystém a přiřazen určeným provozům uvnitř nemocnice. Mluvíme pak o IS označovaném také jako distribuovaný IS. Tyto jednotlivé subsystémy pak běží odděleně na vlastních serverech, uživatel pak pracuje podle oprávnění v určitém subsystému, kde zpracovává pouze konkrétní úlohu. Obvykle tento systém pracuje na modelu klient – server, kdy klientem se rozumí jednotlivé počítače či stanice. Velmi důležitou složkou v kompozitní architektuře je komunikace mezi subsystémy, ta je prováděna buď každý s každým, případně ji obstarává odpovědný server. Jádrem NIS je potom omezeno jen na správu jednotlivých číselníků a zajištění určitých systémových funkcí jako je záloha dat, monitorování funkcí subsystému atp. Možností také je zaměnit systémové jádro modulem systémové integrace. (5)

1.5.2 Integrace aplikací NIS

Jednou z hlavních priorit NIS je, aby byl homogenním a plně integrovaným nástrojem, který bude plně pokrývat veškeré potřeby jeho uživatelů. To, zda-li zvládne uspokojovat všechny požadavky, jinak řečeno celková kvalita a úroveň NIS, jde ruku v ruce s informačním a datovým propojením jednotlivých modulů a subsystémů, nebo-li celkovou integrací systému. Na integraci lze pohlížet z těchto hledisek - datová integrace, prezentační integrace, funkční integrace. (5)

Datová integrace – představuje situaci, kdy pořízená data jednou aplikací jsou automaticky dostupná všem ostatním aplikacím v systému. Lze tedy říci, že pokud jsou data zadána správně, tak mohou být sdílena mezi všemi částmi systému, které si o ně požádají. (5)

Prezentační integrace – zajišťuje, že uživatel systému může zobrazovat vložená data, aniž by ho nějakým způsobem omezovalo místo nebo čas jejich pořízení. Stará se také o konzistentnost a adekvátnost uložených dat, aby všude představovala stejnou nesenou hodnotu. (5)

Funkční integrace – pomocí ní může oprávněný uživatel pracovat s jakoukoliv částí systému a využívat jeho služby a funkce v rámci jednoho komunikačního rozhraní. (5)

Jednotlivé části NIS spolu pak komunikují přímo pomocí aplikačních programových interfaců (API), nebo si vzájemně vyměňují zprávy. Ve druhém zmiňovaném případě komunikace je nezbytné, aby zprávy byly v pořádku a ve správném pořadí doručeny na své místo určení a neporušovaly zvolený komunikační protokol, který zajišťuje jejich správný formát. U heterogenních NIS se tedy nevyhneme použití některého z těchto standardních protokolů. V dnešní době je nejvíce využíván standard HL7 (Health Level Seven). V Česku byl pro tyto potřeby vyvinut standard DS MZ ČR, jímž se řídí většina současných vývojářů NIS. Co se týče obrazové informace a jejího přenosu, celosvětově se používá standard DICOM (Digital Imaging and Communication in Medicine). (5)

1.6 Datové a komunikační standardy

Komunikace mezi jednotlivými prvky systému, případně mezi celými systémy, je základním prvkem pro správné fungování a celistvost navrhovaných architektur. Při návrhu bychom se měli snažit o co nejmenší přímou výměnu dat mezi databázemi jednotlivých aplikací pomocí API a naopak se soustředit na výměnu informací pomocí přenosu zpráv. Formát přenosu a forma zpráv pak musí být známa pro obě komunikující strany, jinak řečeno komunikační a datový standard musí být pro všechny účastníky komunikace známý a stejný. V historii již ve zdravotnictví vzniklo několik, ať již dílčích či národních standardů, což komplikuje požadovanou jednotnost a vizi pouze jednoho komplexního standardu, kterým se budou řídit všichni. V Česku byl vyvinut a dodnes se využívá Český datový standard DS a z mezinárodních standardů se do povědomí dostal HL7 a DICOM. (5)

1.6.1 Český datový standard DS

Jinak známý také jako DASTA se v několika postupných verzích využívá již více než deset let pro předávání informací mezi zdravotnickými informačními systémy. Je použit ve všech významnějších zdravotních informačních systémech České a Slovenské republiky. Jeho vývoj započal na popud Ministerstva zdravotnictví, lékařských fakult a firem vytvářejících IS pro zdravotnictví v roce 1992 po zvážení všech možných variant, jak zajistit vhodný datový standard pro naše území. V průběhu času přicházely lepší verze standardu DASTA, které přinášely lepší funkčnosti pro komunikaci. Aktuálně jsou nejrozšířenější verze DS 3 a DS 4. (5)

Vzhledem k tomu, že se informace mezi systémy přenáší v podobě zpráv, je tento koncept použit i v tomto standardu. Zde je zpráva rozdělena na jednotlivé bloky, kdy každý blok následně obsahuje určitý druh informace např.: adresa, identifikace pacienta atd. Bloky rozdělujeme na povinné a nepovinné, je jejich definováno několik stovek a mnoho jich umožňuje vnořit do sebe další bloky zpráv. (5)

Tento standard má tu výhodu, že byl tvořen v lokálních podmínkách pro lokální potřeby na našem území, čímž plně pokrývá vše, co se od něj očekává. Bohužel s postupem času a požadavkem na “celosvětovou integraci” zdravotnických systémů se právě jeho subjektivita stává jeho nevýhodou. (5)

1.6.2 Standard HL7

HL7 (Health Level Seven) je aktuálně nejpoužívanější datový a komunikační standard ve zdravotnictví vůbec. Byl vyvinut v USA a velmi hojně se užívá, jako nástroj pro integraci heterogenních informačních systémů ve zdravotnictví. Název souvisí s referenčním modelem ISO OSI, jehož sedmou vrstvou striktně využívá a ostatní jsou pro něj pouze podpůrné. (5)

Během doby opět docházelo ke zlepšování a vytváření novějších verzí tohoto standardu. Ve verzi dvě se začlenila podpora pracovních procesů v lůžkových zdravotnických

zařízeních a další rozšíření přicházela jako upgrady této verze. V dnešní době se využívá verze tři, která se soustředí především na podporu všech procesů a postupů ve zdravotnictví, což je opět o mnoho větší rozšíření než v předchozí druhé verzi. (5)

1.7 Datový sklad

Pod pojmem datový sklad si zde představme komplexní systém datového skladu a nikoliv pouze část tohoto systému pro ukládání dat nesoucí stejnojmenný název datový sklad, tyto pojmy jsou často zaměňovány a při špatném pochopení by další informace nedávaly smysl. (6)

„Datový sklad je systém, který umožňuje shromažďovat, organizovat, uchovávat a sdílet historická data. Zahrnuje použitá data pocházející z provozních systémů, které data zachytávají a používají v kontextu své funkce.“ (6, s. 35)

Tvorba datového skladu se obvykle plánuje pro celý podnik na základě identifikace jeho požadavků, potřeb a obecných technických požadavků. Může se však stát, že bude datový sklad zaměřen na určitou část oboru činností podniku, např. finance, marketing, ale stále musí vycházet v souladu s použitelností pro podnik a IT. (6)

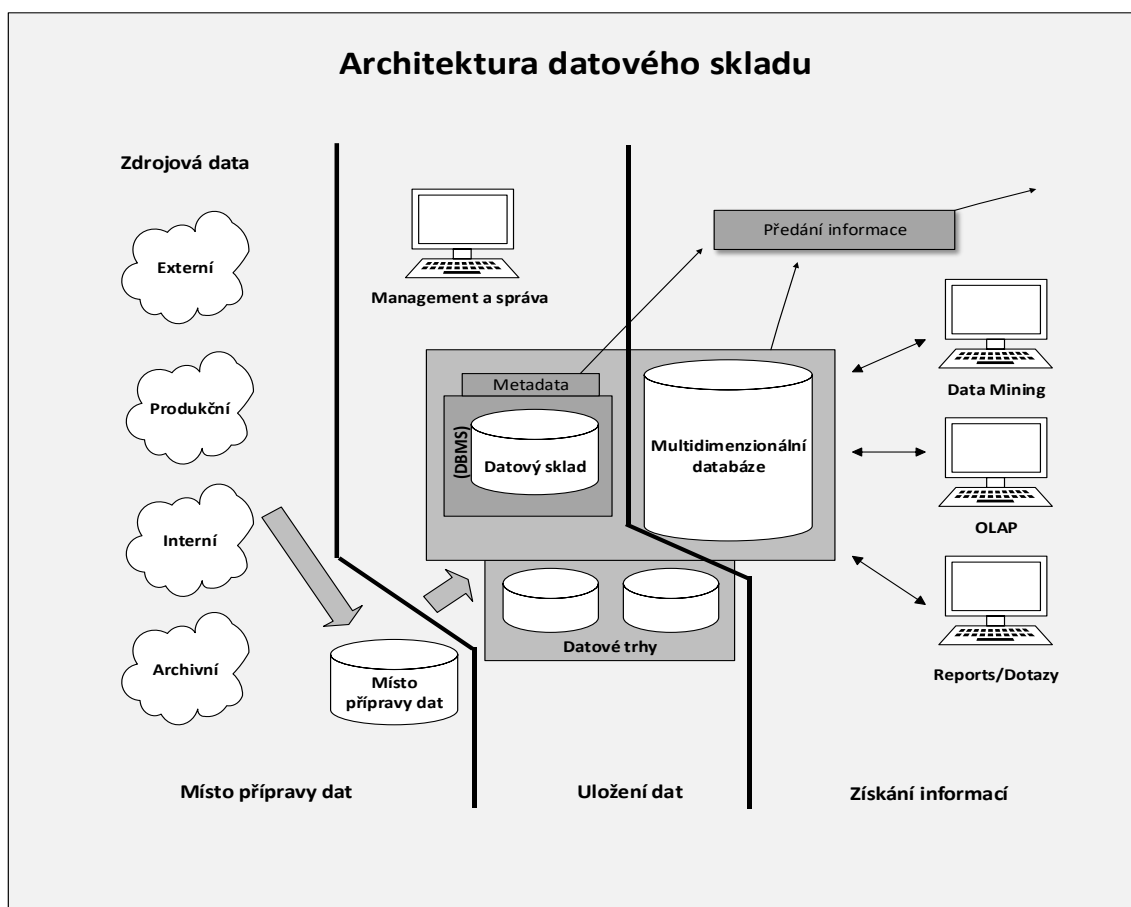
1.7.1 Systém datového skladu

Celková architektura systému pro uložení dat, jejich zprávu a následné vyhodnocení, nebo-li jinak řečeno datový sklad, se může rozdělit na tři hlavní části - získání dat, uložení dat a získávání informací. (6)

- **Získání dat** – data, která chceme uchovávat, se získávají z různých a obvykle více zdrojových systémů; tato data lze dále dělit na produkční, interní, externí a archivní. Většinou je naším cílem uchovávat data, která se týkají subjektu či oblasti našeho zájmu a chceme o nich zjistit další možné informace. Po získání těchto dat a před další fází uložení je nutné tato data transformovat a upravit do vhodné podoby pro uložení do datového skladu, tzn. odstranit anomálie, přidat chybějící hodnoty, změnit formáty, případně strukturu dat. (7 prezentace)
- **Uložení dat** – v této fázi se upravená data ve vhodném formátu ukládají do datových skladů. Podstatou datového skladu je sběr dat za co nejdelší časové období, do skladu se tedy data vesměs pouze přidávají a téměř nikdy se

neodstraňují. Pro popis uložených dat, procesů, funkcí, které na ně byly použity, zdrojů, ze kterých se data vzala, i pro informace o struktuře datových skladů se využívají metadata, jinými slovy data o datech - obsahují všechny potřebné informace pro práci s datovými sklady a pro jejich případné další zpracování. Metadata jsou také využívána částí Managementu a správy, která se nachází v této oblasti uložení dat. Ta má za úkol dohlížet na postupy a průběh práce uvnitř celé této architektury. (7 prezentace)

- **Získávání informací** – zde se na základě různě složitých dotazů, pomocí metod Data Miningu, případně různých druhů OLAP analýz, získávají z uložených dat informace, které zprvu v uložených kvantech dat nebyly patrné a právě díky těmto metodám je můžeme extrahovat a použít je pro další rozhodování. Použití jednotlivých metod a jejich různých stupňů složitosti závisí především na zkušenostech uživatele, který se snaží získat požadované informace. (7)



Obrázek č. 1: Architektura datového skladu (Vlastní zpracování)

1.8 Business intelligence

Pojem business intelligence (BI) je pevně svázán se systémem datového skladu, který je popsán výše. Je zcela zjevné, jak jsou tyto dva celky závislé a v jaké fázi dochází k transformaci dat na informaci hodnotnou pro koncového uživatele. Je to právě BI, jenž je hodnotná pro podnik, který podle ní činí důležitá rozhodnutí. Stejně, jako tomu bylo u návrhu datového skladu, i zde je potřeba zvážit účel, obor a typ BI, aby se v podniku nevytvářely stále stejné sestavy a analýzy i přesto, že nejsou nikterak užitečné. Při správném zvážení a výběru typu analýzy BI a metody poskytování výstupu se výrazně zlepší efektivita celého systému a dosáhneme požadovaných výsledků, kvůli kterým byl celý systém zhotoven. Pro lepší pochopení funkce a činnosti business intelligence v podniku přikládám konkrétní definici toho pojmu. (6)

„Business intelligence je zastřešující termín, který se vztahuje ke znalostem, procesům, technologiím, aplikacím a postupům, které usnadňují podnikové rozhodování.“ (6, s. 33)

Velkým přínosem tohoto konceptu je především rychlost a případná jednoduchost, se kterou můžeme informace získávat. Ještě nedávno zaměstnala operace na vytvoření určité sestavy několik lidí z více oddělení, doba provádění této činnosti se podle složitosti pohybovala v řádech dnů až týdnů, během nichž docházelo k upřesňování a lepší specifikaci požadavků mezi stranou tazatele a lidmi zpracovávající jeho zadání. Dnes pomocí nástrojů BI může tuto sestavu vytvořit běžný uživatel během několika minut, celý proces se tak mnohonásobně zrychlil společně s tokem informací. (6)

1.9 Nástroje InterSystems

V této kapitole budou rozebrány jednotlivé nástroje společnosti InterSystems, jenž jsou později využity pro návrh řešení mnou zvoleného informačního systému.

1.9.1 Caché

Jedná se o vysoce výkonnou objektovou databázi, databázový stroj, který nabízí rozmanitou sadu služeb pro zhotovení komplexních systémů na správu databází. Díky těmto službám Caché umožňuje kompletně spravovat objekty a relační databáze. Tento moderní nástroj překonává omezení relačního modelu databází, ale zároveň zachovává podporu mnoha aplikací a nástrojům, které jsou založeny na SQL. V Caché lze

přístupovat k datům jak objektově orientovaným způsobem, tak pomocí tradičních SQL dotazů, kdy jsou data uložena ve formě relační tabulky. Mělo by se jednat o jeden z nejrychlejších, nejspolehlivějších a nejlépe škálovatelných nástrojů tohoto typu, který je v dnešní době dostupný na trhu. Právě velká rozmanitost z hlediska škálovatelnosti je velkou výhodou, je jednoduché nastavit databázi pro využívání pouze jedním uživatelem, stejně tak jako používání této databáze pro aplikační platformu rozsáhlé zdravotnické sítě, která provozuje stovky aplikací důležitých pro pacienta a má desetitisíce klientských stanic. (4)

1.9.2 Ensemble

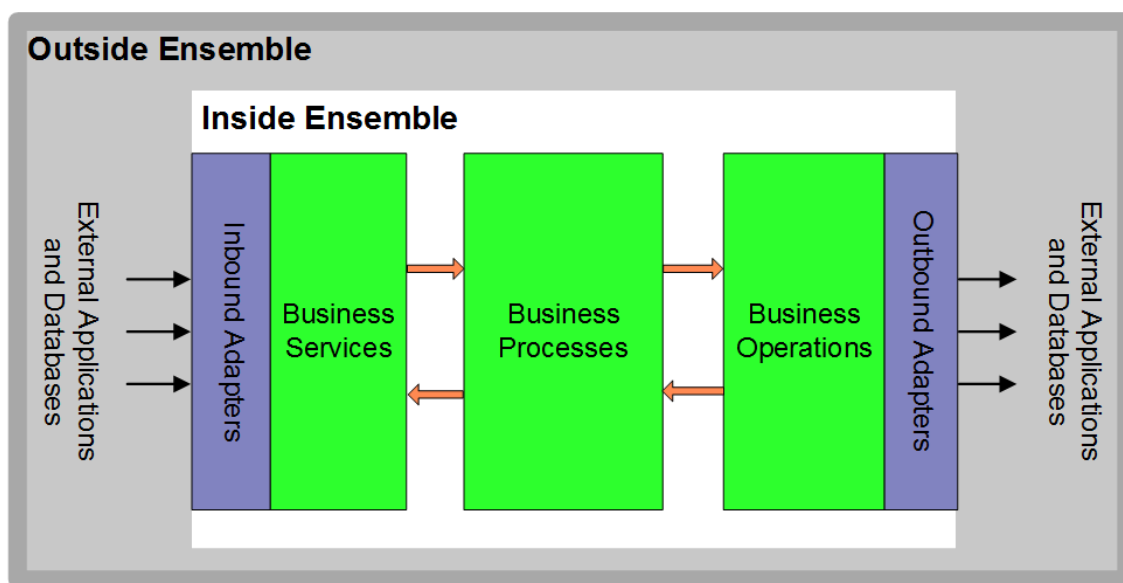
Představuje robustní a výkonný integrační nástroj vhodný pro vývoj a propojení jednotlivých aplikací. Vzhledem ke kompaktnosti tohoto produktu se značně zjednodušuje celkový proces integrace a poskytuje nová řešení, která se tvoří rychleji a lépe se spravují. Součástí Ensemblu je také již zmiňovaný databázový systém Caché, kdy Ensemble stojí na jeho samotném vrcholu a dokáže tedy nabídnout všechny jeho služby, ke kterým navíc přidává také svoje specifické funkce. Společně s několika univerzálními programovacími jazyky, které jsou součástí Caché, lze vytvářet nová vlastní nastavení, čímž je proces integrace aplikací mnohem pohodlnější. (4)

Tok komunikace řídí v Ensemblu Produkce, která se dělí na tři hlavní části - Business services (Služby), Business processes (Procesy) a Business operations (Operace). Všechny tyto části mají svoje specifické zprávy, které nesou informace o dění uvnitř produkce. Jednotlivé zprávy pak lze zobrazovat přes prostředí Management portálu v Ensemblu obsaženém. (4)

Business services (Služby) – stojí na počátku procesu zpracování dat v Ensemblu. Pomocí vstupního adaptéru, který je obvykle přidružen a jenž lze nastavit případně nově vytvořit, získávají data z okolí. Tato data jsou pak nadále zpracovávána v dalších částech produkce. (4)

Business processes (Procesy) – starají se o příjem dat ze služeb a jejich následné zpracování nebo přeposlání do dalších operací. Po dokončení mohou být data předána do poslední části produkce. (4)

Business operations (Operace) – nacházejí se na konci celého procesu a přijímají data od předchozích částí Produkce, která pak mohou zpracovat či distribuovat dalším subjektům mimo Ensemble. Opět obvykle obsahují výstupní adaptéry, které jsou nastavitelné, čímž ulehčují propojení s externími zdroji. (4)



Obrázek č. 2: Schéma komponent uvnitř Ensemblu (Zdroj: 4)

1.9.3 DeepSee

S využitím tohoto nástroje je možné vnést do našich projektů prvky Business Intelligence (BI), což nám umožňuje tvořit sofistikované dotazy nad daty, které zpracováváme a vytvářet přehledné výstupy, které jsou srozumitelné pro uživatele. Na rozdíl od běžných nástrojů BI dokáže DeepSee pracovat s aktuálními právě používanými daty v našem projektu, což opět zlepšuje kvalitu výsledku. Mezi jeho stěžejní části patří Architect, Analyzer a User Portal. (4)

Architekt – pomocí Architectu jdou vytvářet datové kostky, u nichž dále specifikujeme jednotlivé dimenze a hierarchie, což nám umožňuje lépe vyspecifikovat potřebná data.

Pokud chceme ještě více zúžit, vyspecifikovat konkrétní data, můžeme se zaměřit pouze na určitou část kostky např. pomocí nástroje Subject Areas. (4)

Analyzer – umožňuje vytvářet kontingenční tabulky, které zobrazují agregovaná data podle toho, jakým způsobem si uživatel navolí omezení pro data, jenž ho zajímají. Vše lze exportovat do jiných aplikací např. Excelu, případně využít User Portal pro zobrazení grafu. (4)

User Portal – je určen pro finální zobrazení výsledků dat pro koncového uživatele. Je zde možné vytvářet přehledné prezentace výsledků ve formě grafických objektů, které mohou zpracovávat kontingenční tabulky z Analyzáru a dále je zkoumat. Vše lze navíc sdílet mezi dalšími uživateli daného User Portalu pro stejnou instanci Caché, kteří do něj mají přístup, mohou si navzájem přiřazovat úkoly, sdílet informace, nebo zanechávat zpětné vazby, což urychluje a usnadňuje práci při vývoji. (4)

2 ANALÝZA SOUČASNÉHO STAVU

V této části mojí práce se budu zabývat popisem aktuálního stavu ve Fakultní nemocnici Brno, budou zde zahrnuty jak vnitřní, tak vnější vlivy na tuto organizaci, které na ni působí ať již pozitivním či negativním způsobem. Pro analýzu jsem se rozhodl využít modely SLEPT, 7S a výsledky z těchto dvou analýz zahrnu i do finální analýzy SWOT. Především při analýze vnitřního prostředí se zaměřím na důkladný popis informačních systémů, které aktuálně nemocnice používá, abych mohl následně lépe přizpůsobit mnou navrhovanou část informačního systému.

2.1 SLEPT analýza

Tato sekce bude zaměřena na tvorbu analýzy SLEPT, která se zaměřuje na zmapování okolního prostředí nemocnice a snaží se zachytit aktuální stav působících okolních faktorů, jež budou vypsány níže a ideálně predikovat jejich budoucí vývoj.

2.1.1 Sociální faktory

Oblast sociálních faktorů je pro tuto organizaci stěžejní, zejména demografické rozložení společnosti totiž ovlivňuje hned několik dalších oblastí, které jsou navzájem provázány. Skladba a početnost obyvatelstva má vliv na velikost výdajů spojených s ošetřováním a péčí o pacienty, stejně tak působí na kapacitní limity nemocnici a v neposlední řadě na vytíženost lékařů a personálu pracujícího v této organizaci, což může opět vést ke zvyšování nákladů. Vezmeme-li v úvahu počet obyvatel a jejich složení, které můžeme vidět v tabulce níže, v městě Brně, kde se nachází i hlavní pole působnosti FN Brno, vidíme, že za posledních několik let dochází k úbytku hlavní pracovně aktivní části obyvatelstva a naopak lehce se zvyšují počty dětí a seniorů, kteří využívají služeb nemocnice nejčastěji. Stejně tak je vidět pomalu rostoucí počet obyvatel města, vše odpovídá obecným předpokladům o zvyšování a stárnutí populace České republiky.

Tab. 1: Rozložení a počty obyvatel v Brně v posledních letech (Vlastní zpracování)

| Rok | Počet obyvatel podle věkových skupin | | | Celkový počet obyvatel | Průměrný věk |
|------|--------------------------------------|---------|--------|------------------------|--------------|
| | 0 - 14 | 15 - 64 | 65 + | | |
| 2011 | 51 757 | 257 397 | 69 811 | 378 965 | 42,2 |
| 2012 | 52 615 | 254 105 | 71 607 | 378 327 | 42,3 |
| 2013 | 53 479 | 251 000 | 73 029 | 377 508 | 42,5 |
| 2014 | 54 492 | 248 709 | 74 239 | 377 440 | 42,6 |
| 2015 | 55 325 | 246 583 | 75 120 | 377 028 | 42,7 |
| 2016 | 56 413 | 245 178 | 76 382 | 377 973 | 42,8 |
| 2017 | 57 598 | 244 455 | 77 474 | 379 527 | 42,8 |
| 2018 | 58 672 | 243 614 | 78 395 | 380 681 | 42,8 |

Pro porovnání určité závislosti mezi skladbou, počtem obyvatel a počtem ošetření v nemocnici za rok, je přiložena následující tabulka, kde je vidět zřetelný meziroční nárůst provedených ošetření pacientů FN Brno.

Tab. 2: Počet ambulantních ošetření ve FN Brno za jednotlivé roky (Vlastní zpracování)

| Počet ambulantních vyšetření ve FN Brno | | | | | | |
|---|-----------|-----------|-----------|-----------|-----------|-----------|
| Rok | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 |
| Ambulantní vyšetření | 1 023 639 | 1 036 358 | 1 062 961 | 1 070 696 | 1 077 073 | 1 086 799 |

2.1.2 Legislativní faktory

Činnost FN Brno je ovlivňována především platnou právní legislativou týkající se zdravotnictví, jíž spravuje Ministerstvo zdravotnictví ČR, které je zároveň zřizovatelem celé nemocnice, případně vyhláškami Evropské Unie a dalšími aktéry v politických kruzích. Zároveň musí respektovat ať již určité články České ústavy či Listinu základních práv a svobod. Co se týče technologického směru, je důležité uvést národní strategii pro elektronické zdravotnictví. Pro představu o cílech tohoto projektu uvedu citaci ze strategického plánu Ministerstva zdravotnictví v oblasti elektronizace zdravotnictví: „ČR deklaruje jasný záměr postupně sladit vzájemně nekoordinované aktivity státu, samosprávy, komerčního sektoru, zdravotních pojišťoven a poskytovatelů zdravotních služeb směrem k naplnění vize Národní strategie elektronického zdravotnictví a k

vytvoření interoperabilního důvěryhodného prostředí pro všechny realizátory a uživatele služeb elektronického zdravotnictví.“ (8) Toto rozhodnutí poskytuje možnou predikci budoucího vývoje v oblasti zdravotnictví a možnost pro uplatnění nových řešení.

2.1.3 Ekonomické faktory

Již bylo zmíněno, že zřizovatelem FN Brno je MZ ČR, z čehož plyne, že se jedná o státní příspěvkovou organizaci, jejímž hlavním zdrojem financování jsou příjmy za poskytnutou léčbu od zdravotních pojišťoven, které mohou doplňovat různé granty, dotace, případně sponzorské dary. Z velké míry však mohou mít na ekonomickou situaci nemocnice vliv uvedené legislativní či politické faktory. (9)

2.1.4 Politické faktory

V oblasti politických faktorů, které mají vliv na nemocnici, vytvořilo MZ ČR ve spolupráci se Státním zdravotním ústavem a Kanceláří Světové zdravotnické organizace (WHO) v České republice důležitý dokument „Zdraví 2020“, který udává obecný směr, jímž by se mělo zdravotnictví, jeho podpora a vývoj v nejbližších letech ubírat, ať už po stránce prevence nemocí a ochrany zdraví, globalizace zdravotnictví a jeho dostupnosti pro každého, či způsoby financování těchto kroků. (10)

2.1.5 Technologické faktory

Z legislativních faktorů vyplývá snaha o elektronizaci zdravotnictví, pro nemocnici to tedy znamená zajištění komunikace mezi všemi systémy a zařízeními, které spravuje a případné přidání do integrovaného informačního systému, který je v nemocnici používán. Zároveň ve FN Brno neustále probíhá proces modernizace a snaha o zavádění nejnovějších technologií. Pro tyto účely bylo zřízeno Centrum informatiky v nemocnici, kde zaměstnanci, jakožto zkušení odborníci na danou problematiku, pomáhají v rozhodování, jakým způsobem lze určité činnosti v tomto specifickém prostředí provést, sledují vývoj trendů z této oblasti, zvažují jejich přínosy pro nemocnici a případně se starají nebo koordinují jejich implementaci. (11)

2.2 Analýza 7S

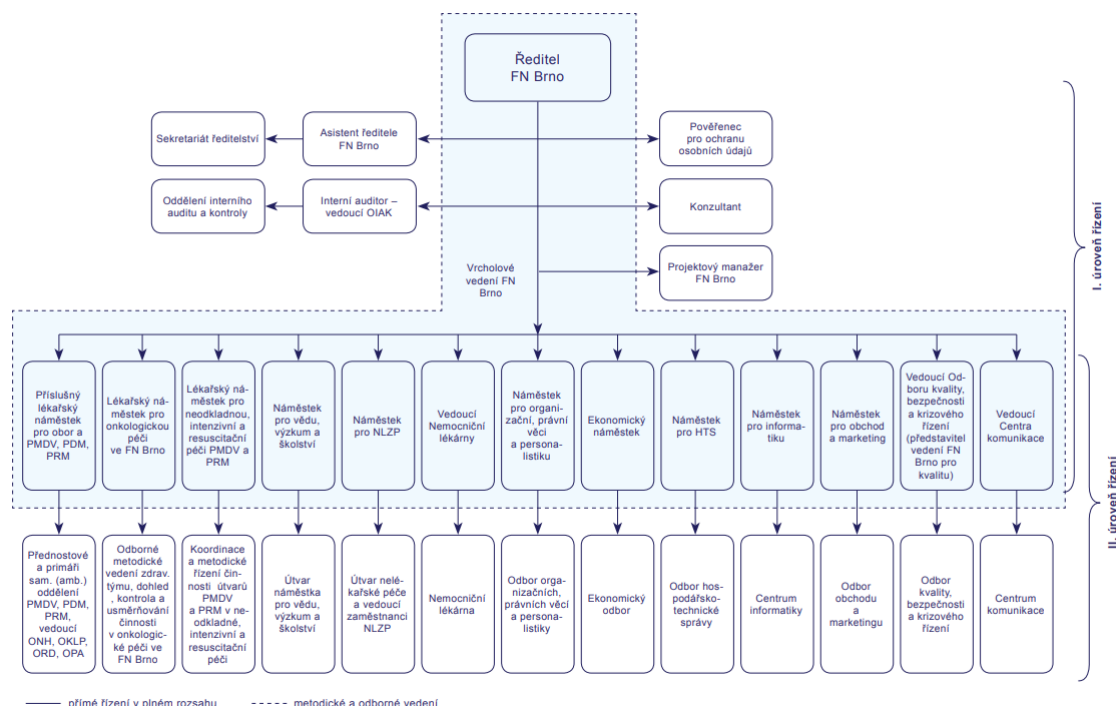
V další sekci bude provedena analýza 7S, nebo-li také McKinseyho model. Tento druh analýzy je zaměřen především na vnitřní prostředí podniku, kterým je v tomto případě stále FN Brno.

2.2.1 Strategie

Strategie ve FN Brno, jakožto druhého největšího poskytovatele zdravotních služeb v České republice, je jasná - péče o pacienta a jeho spokojenost je na prvním místě, s čímž souvisí i poskytnutí komplexní péče ve všech medicínských disciplínách s důrazem na neustálý vývoj v těchto oblastech. Součástí této strategie je i zavádění nových poznatků vědy a technologií, do tohoto segmentu patří právě modernizace ICT struktur uvnitř nemocnice spolu se vznikem mnohých nových projektů spojených právě s elektronizací zdravotnictví, např.: jednotný patientský záznam Master Patient Index (MPI), či Electronic Health Record (EHR), jenž zprostředkovává uložení a manipulaci se zdravotnickou dokumentací pacienta.

2.2.2 Struktura

V současné době je ve vrcholném vedení FN Brno nastolena liniově-štabní organizační struktura, na jejímž vrcholu stojí ředitel, kterému se zodpovídají všechny ostatní části této struktury. Na každém oddělení je pak na nejvyšším postu náměstek daného oddělení. V nižších částech této struktury se však tato ucelená forma rozpadá spíše na maticovou, což není zcela vyhovující, obzvláště, pokud přihlédneme k velikosti celé organizace, vede tato změna k potížím při controllingu, personálních změnách a působí také na ekonomickou oblast. Na následujícím obrázku můžeme vidět nejvyšší prvky této struktury spolu s rozdělením na nejzákladnější části.



Obrázek 1: Schéma organizační struktury FN Brno (Zdroj 11)

2.2.3 Systémy

V tomto bodě analýzy se promítnou hlavní používané interní systémy ve FN Brno. Jak již bylo zmíněno v předchozích odstavcích, nemocnice využívá nemalé množství systémů pro různé druhy činností, od systémů pro záznam docházky, přes ekonomické, až po systémy pro správu pacientů, případně mnohé další. Z teoretické části a zároveň ze SLEPT analýzy víme, že pro správnou funkci a celkový chod organizace je třeba tyto systémy mezi sebou propojit, aby mohla být data mezi nimi přístupná. Ve FN Brno k této potřebě využívají technologie společnosti InterSystems, které zajišťují sjednocení do integrační platformy a tím zefektivňují tok dat uvnitř organizace.



Obrázek 2: Propojení informačních systémů s integrační platformou (Vlastní zpracování)

ENIS

Tento systém, který je postaven a využívá základů z integrační platformy od firmy InterSystems, je jedním z hlavních a aktuálně poslední zřizovaný systém ve FN Brno. ENIS pomáhá integrovat řízení procesů uvnitř nemocnice, které také zjednodušuje. Zároveň zlepšuje práci se zdravotnickou dokumentací a zkvalitňuje vykazování pro zdravotní pojišťovny. Další výhodou je, že toto řešení je využitelné i pro další nemocnice v ČR.

AMIS H

Stěžejním systémem pro FN Brno je pak AMIS H, v současné době již převeden na novější verzi AMIS HD, která však stále plně nekoresponduje s aktuálními trendy a moderními IS. Nemocnice si však doposud s tímto klinickým systémem pro podporu běžných procesů lékařského a ošetrovatelského personálu vystačila navzdory tomu, že jeho rychlost je značně nedostačující aktuálním potřebám. Zmiňovaný přechod na novou

verzi nijak zvlášť tento problém nevyřešil, jednalo se především o vizuální stránku věci, tzn. převod z terminálového zobrazení na GUI (grafické uživatelské rozhraní).

MPA

Posledním ze systémů, které jsem se rozhodl více rozvést, je systém MPA (Medical Proces Assistant). V porovnání se systémem AMIS H umožňuje automatizovat jednotlivé procesy při práci s pacientem, což šetří čas a zjednodušuje práci lékařům a taktéž pacientům, kterým se zkracují čekací doby, případně zrychluje příjem urgentních případů. Jako další výhodu je třeba uvést zvýšení bezpečnosti a ochrany údajů pacientů, kterou tento systém přináší. Dokončování implementace tohoto systému pak probíhalo v roce 2008 a byl to první takovýto IS svého druhu v ČR.

Ostatní systémy

V nemocnici se pracuje s velkým množstvím dalších systémů, z nichž jsou některé uvedeny na obrázku výše, avšak vzhledem k rozsahu a zaměření mé práce mi nepříjdou ve srovnání s uvedenými, i přes svoji jistou roli v této organizaci, natolik významné, aby bylo nutné je blíže specifikovat, proto je ve své práci neuvádím.

2.2.4 Styl

Pro FN Brno je typický demokratický styl, kdy svůj hlas mají i jednotliví zaměstnanci. Celá organizace se pak snaží o co největší jednotnost a kompaktnost, spolu s důrazem na rychlost jednání, efektivnost prováděných akcí, důkladnost a profesionalitu. Tento přístup by měl být samozřejmý, jedná se přeci o nemocnici, kde není výjimkou boj o holý život pacienta a většina návštěv této organizace má přímý vliv na naše zdraví, nicméně ve FN Brno všemi těmito vlastnostmi a mnohými dalšími disponují a ke všemu přidávají příjemné a slušné vystupování, což jen podtrhuje celkový dojem, kterým tato organizace působí.

2.2.5 Sdílené hodnoty

Pacient, případně klient na prvním místě, tak zní jedna ze základních hodnot, kterou ctí ve FN Brno, spolu s dalšími hodnotami, jako je jednání zaměstnance s ohledem na zájmy a dobré jméno nemocnice i pacienta a také s dodržováním obecně známých mravních

principů, utvářejí pozitivní a vlídnou firemní kulturu, která koresponduje s mottem nemocnice, jenž zní: „Žijeme pro Vaše zdraví“.

2.2.6 Spolupracovníci

Vzhledem k velikosti FN Brno a jejímu významu je zřejmé, že zaměstnanci jsou obvykle experti ve svém oboru. Avšak i přesto, že si uvědomují svoji významnost a nenahraditelnost, plní misi a vizi této společnosti s maximálním nasazením. Právě nedostatek nových profesionálů v oboru je jednou z příčin, proč jsou tyto práce tak náročné. Proto také FN Brno spolupracuje s Masarykovou univerzitou v Brně, jíž poskytuje vzdělávací prostory a cenné poznatky z praxe, čímž pomáhá vytvářet novou vzdělanou a perspektivní generaci pracovníků v oboru, z nichž mohou být její potencionální zaměstnanci a mohou tak ulevit těm stávajícím, kteří jsou enormně vytížení.

2.2.7 Schopnosti

Z předchozích bodů analýzy je patrné, že schopnosti zaměstnanců a organizace samotné jsou na velmi vysoké úrovni, odbornost a preciznost prováděných prací může být mnohdy rozhodující. Zaměstnanci této organizace však často potřebují širší spektrum znalostí i z jiných oborů než je lékařství, právě proto nemocnice klade důraz i na časté proškolení svých zaměstnanců, aby stále udržovali krok s dobou a nebylo jim cizí např.: ovládání nových přístrojů či prvků ICT a dále pak práce s informačními systémy nemocnice. Zdokonalování schopností ve všech směrech pak přispívá k větší spokojenosti zaměstnanců a efektivnějšímu pracovnímu nasazení.

2.3 SWOT analýza

Zde bude provedena SWOT analýza, která nám, jako jedna ze základních metod strategické analýzy, poskytne finální souhrn informací a vyhodnocení poznatků z předešlých analýz SLEPT a 7S a může nám tak pomoci v rozhodování o budoucí strategii FN Brno. Samotná SWOT analýza je založena na mřížce, v níž figurují silné a slabé stránky, spolu s příležitostmi a hrozbami pro organizaci, tyto sekce jsou uvnitř

jednotlivých polí mřížky, zároveň jsou zde záhlaví sloupců a řádků, jenž vše oddělují na interní a externí analýzu a zároveň kladné a záporné stránky. Vše je názorně vidět na následujícím obrázku.

Tab. 3: SWOT analýza (Vlastní zpracování)

| | Kladné stránky | Záporné stránky |
|------------------------|---|--|
| Vnitřní analýza | Silné stránky | Slabé stránky |
| | <ul style="list-style-type: none"> • Vlastní IT tým složený z odborníků • Velikost FN Brno • Podpora od státu a MSZ • Orientace nemocnice na moderní technologie | <ul style="list-style-type: none"> • Zastaralý IS • Rychlost přenášení dat mezi systémy • Nejednotnost používaných systémů |
| Vnější analýza | Příležitosti | Hrozby |
| | <ul style="list-style-type: none"> • Podpora a snaha o modernizace ve FN Brno • Možnost financování z dotací EU a jiných organizací • Podpora programů pro elektronizaci a integraci ve zdravotnictví • Kompletní integrace všech systémů | <ul style="list-style-type: none"> • Vytíženost IT týmu i samotných lékařů • Odstoupení od snahy o elektronizaci zdravotnictví • Ukončení dotačních programů • Přílišná investice do zastaralého IS • Změna strategie nemocnice |

2.4 Analýza požadavků zadavatele

V této sekci budou uvedeny jednotlivé požadavky pro funkčnost modulu žádankového portálu, jenž byly vydefinovány v zadavatelské dokumentaci, kterou pomáhala sestavit FN Brno pro plné uspokojení všech jejích potřeb. Požadavky jsou uvedeny kompletním výčtem, avšak vzhledem k rozsahu této práce podrobněji zpracuji řešení pouze vybraných požadavků, ostatní budou buď řešeny podobným způsobem, či bude nastíněno pouze řešení obecné.

Požadavky, jenž musí být splněny pro plnou funkčnost modulu:

- Požadavky na integraci s integrační platformou nemocnice:
 - Integrace EOP (žádankového portálu) do MPI a EHR přes integrační platformu FN Brno
 - Přebírání společných číselníků z integrační platformy FN Brno
 - Synchronizace centrálního číselníku provedených vyšetření s integrační platformou FN Brno
 - Synchronizace centrálního číselníku provedených vyšetření s integrační platformou FN Brno
 - Zjišťování stavu zpracování požadavků v cílovém pracovišti v integrační platformě FN Brno
 - Získání výsledků žádankového vyšetření z integrační platformy FN Brno
- Funkčnosti samotné žádanky/žádankového portálu:
 - Odesílání elektronické žádanky na zpracování do integrační platformy FN Brno
 - Vytvořit objednávku na různé vyšetření ve flexibilním formuláři bez ohledu na konkrétní cílové pracoviště, druh materiálu a typ odběru
 - K vyšetřením vybraným ve flexibilním formuláři zobrazit další informace vztahující se k druhu a podmínkám odběru, laboratoři a pacientovi
 - Možnost definovat různé typy šablon flexibilního formuláře pro vytvoření objednávky pro pracoviště a uživatele
 - Rozdělit požadované vyšetření do žádanek podle cílového pracoviště
 - Možnost definovat volný text poznámky k jednotlivým polím objednávky

- Žádanka by měla být jednoznačně identifikovatelná čárovým kódem
 - Kontrolovat správnost zadaných údajů pacienta vzhledem k hodnotám v číselnících
 - Kontrolovat správnost vazeb mezi metodami, případně parametry metod
 - Kontrolovat a upozorňovat na početnosti vyšetření
 - Musí obsahovat možnost editovat žádanku
 - Musí obsahovat možnost tisknout žádanku v tiskové podobě
 - Možnost editace přiřazeného vzorku k žádance
 - Možnost zobrazovat výsledky podle přístupových práv uživatele
 - Musí evidovat informaci o čase odezvy
 - Musí prezentovat výsledky vyšetření
 - Musí zobrazovat přehledy zadaných žádanek
 - Musí auditovat činnosti uživatelů systému
 - Musí integrovat žádanku s evidencí informovaných souhlasů pacienta
- Požadavky na data, jenž musí každá žádanka obsahovat:
 - Žádanka musí obsahovat osobní údaje pacienta a údaje o pojištění
 - Žádanka musí obsahovat diagnózu, případně víc typů diagnóz
 - Žádanka musí obsahovat urgentnost vyšetření
 - Žádanka musí obsahovat údaje žádajícího subjektu
 - Žádanka musí obsahovat údaje o vzorku (pokud je vzorek nutný k vyšetření)
 - Žádanka musí obsahovat požadované vyšetření
 - Žádanka musí obsahovat požadované vyšetření seskupené do logických celků a souborů vyšetření
 - Žádanka musí obsahovat klinicky relevantní údaje pacienta podle druhu požadovaného vyšetření
 - Žádanka musí obsahovat pole pro poznámku
 - K žádance musí být možnost přiložit soubor
 - Požadavky na vzorky přikládání k žádance:

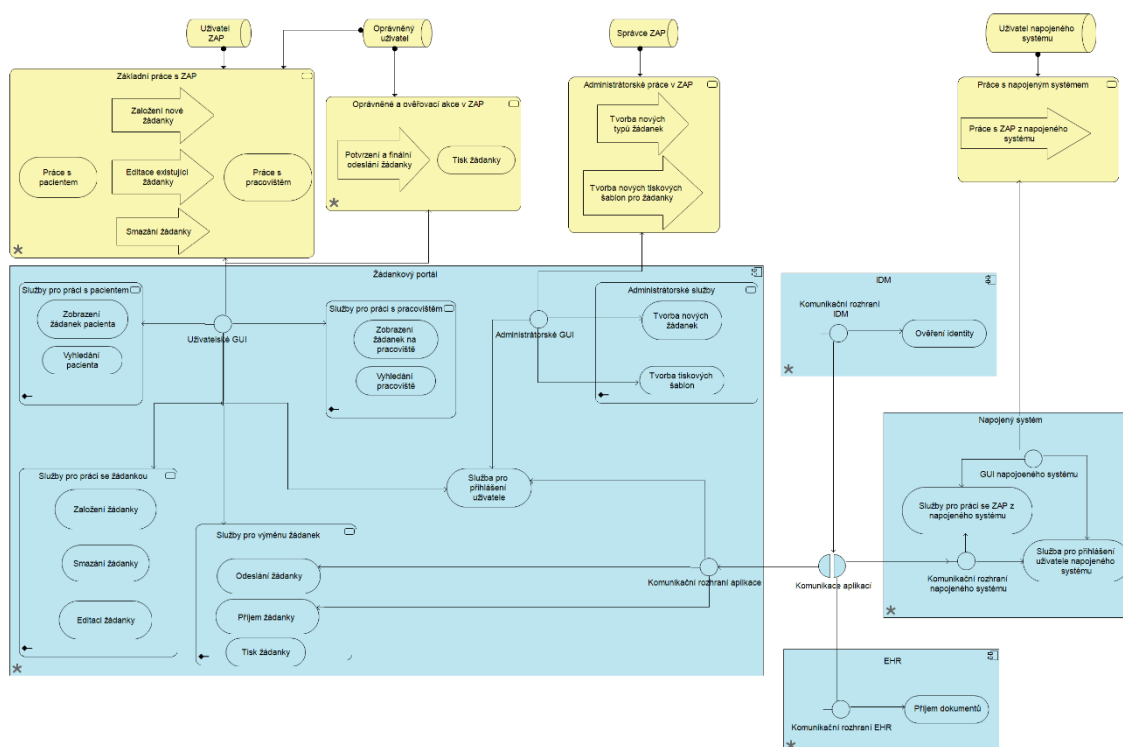
- Vzorek musí obsahovat informace o druhu materiálu, datumu, času odběru, kým byl daný odběr uskutečněn
- Vzorek musí obsahovat podrobnější informace k odběru – podle druhu materiálu a typu vyšetření
- Vzorek musí být jednoznačně identifikován čárovým kódem

3 VLASTNÍ NÁVRHY ŘEŠENÍ

Zde bude popsáno a navrženo celkové fungování a rozložení žádankového portálu. Pro znázornění jsem se rozhodl využít modelovací jazyk ArchiMate, jehož definice nám umožní jak pohled na systém jako celek, tak pohled na jednotlivé dílčí části tohoto systému.

3.1 Architektura žádankového portálu

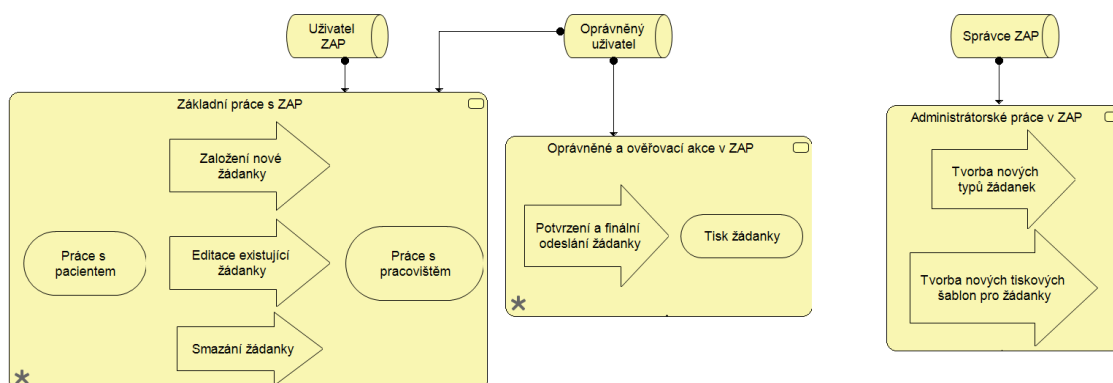
Na následujícím obrázku je znázorněna celková architektura navrhovaného systému pomocí jazyku ArchiMate. Jeho podstatou je pak rozdělení do jednotlivých, vzájemně propojených vrstev. Nejvýše stojí Business vrstva, která je v diagramu zabarvena žlutě, v této vrstvě působí jednotlivé role, ke kterým náleží určité procesy a operace, jenž lze v systému vykonávat. Dále navazuje vrstva aplikační, značena modře, jenž poskytuje propojení s Business vrstvou. V této vrstvě jsou zachyceny určité služby obsluhující především procesy z vyšší vrstvy, zachycena jsou zde také rozhraní pro komunikaci a to jak s uživatelem, tak s jinými aplikacemi či systémy.



Obrázek č. 3: Diagram pro aplikaci ZAP (Vlastní zpracování)

3.2 Business vrstva

V rámci této vrstvy bude popsáno využívání žádankového portálu spolu se specifikací jednotlivých uživatelů a jejich činností, jež budou moci s portálem vykonávat.



Obrázek č. 4: Diagram zobrazující business vrstvu aplikace ZAP (Vlastní zpracování)

Jednotlivé procesy v aplikaci jsou pak přiřazeny uživatelům, kteří je provádějí manuální prací s portálem; tyto procesy jsou z většiny soustředěny právě „uvnitř“ žádankového portálu, na rozdíl od automatizovaných procesů, které využívají nebo jsou vykonávány v napojených či zdrojových systémech. Data pak musejí být vhodně přenášena, aby bylo dosaženo správného fungování portálu.

S žádankovým portálem budou pracovat především tyto uživatelé:

Lékaři, sestry a další pověření pracovníci – jejich práce s touto aplikací bude spočívat především ve vystavování nových žádanek pro pacienty a následně činnosti s tím spojené, jejich přesný výčet je patrný z UseCase diagramu celé aplikace, případně pak z jednotlivých případů užití. Tento druh uživatelů využívá **PU 1 – 9**.

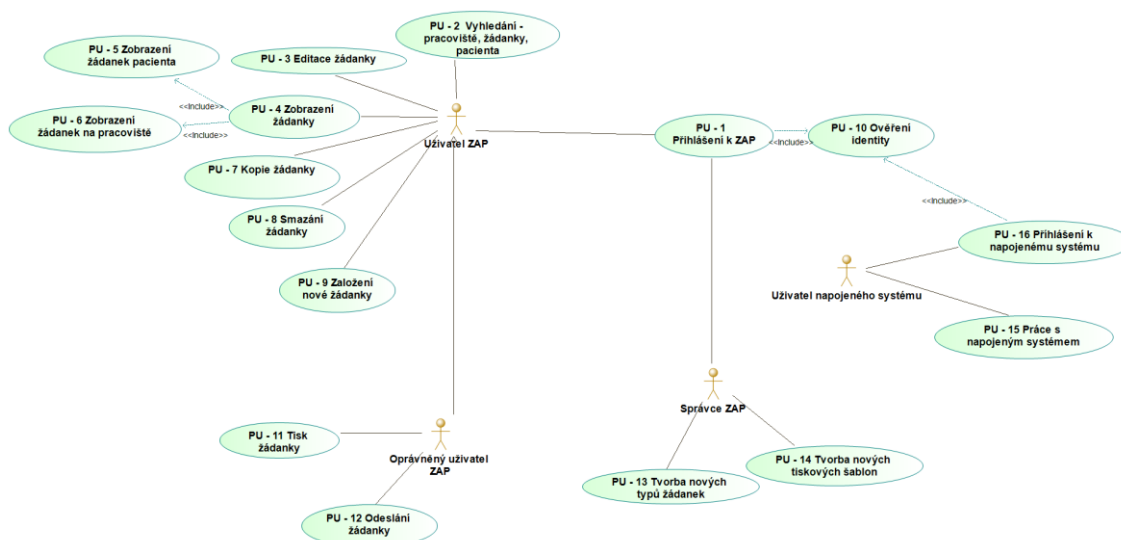
Oprávnění lékaři, sestry a další pověření pracovníci – tyto uživatelé ZAP přejímají veškerá práva od předchozí skupiny a navíc jsou jim umožněny další činnosti s touto aplikací. Především se bude jednat o vedoucí jednotlivých úseků či jiné nadřízené pracovníky, jelikož jednou z jejich hlavních činností bude schvalování vystavených

žádanek a jejich odesílání k plnění; tato činnost vyžaduje vyšší kompetence, jež jsou obsaženy v této skupině uživatelů. Tito uživatelé pak pracují s **PU 1 – 12**.

Správci, administrátoři – úkolem této úzké skupiny uživatelů bude vytváření nových šablon pro žádanky dle požadavků ostatních skupin uživatelů, dále se pak tito uživatelé budou starat o celkový chod aplikace a ostatní věci s tím související, tomu také budou odpovídat jejich práva, která jsou nejvyšší ze všech zmíněných skupin, mohou tedy téměř vše, ale pro účely jejich práce budou využívat hlavní **PU 1, 13 a 14**.

3.2.1 Případy užití

Zde bude poskytnut bližší pohled na aktéry v Business vrstvě, spolu se specifikací jejich činností. Pro obecné znázornění činností bude použit Use Case diagram tvořený podle zásad standardu UML. Následně budou připojeny jednotlivé tabulky, v nichž budou uvedeni konkrétní případy užití, k nimž budou připojeni související aktéři pracující s žádankovým portálem.



Obrázek č. 5: Diagram případů užití pro aplikaci ZAP (Vlastní zpracování)

Tab. 4: Příklad užití pro přihlášení k ZAP (vlastní zpracování)

| | | | |
|-------------------------|--|--------------|--|
| Název: | PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému | | |
| Popis: | Jakýkoliv uživatel se před prací s ZAP musí přihlásit svým jménem a heslem. | | |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Správce ZAP • Uživatel ZAP (lékař, zdravotní sestra) • Oprávněný uživatel • Uživatel napojeného systému (pracovník laboratoře atp.) | | |
| Vedlejší aktéři: | | | |
| Požadavky pro vykonání: | | | |
| Navazující PU: | <ul style="list-style-type: none"> • PU – 10 Ověření identity | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Spustí aplikaci a zadá uživatelské jméno a heslo do příslušných polí aplikace a potvrdí tlačítkem. |
| | 2. | Aplikace | Ověří zadané údaje v součinnosti s ověřením identity, jehož princip je popsán v PU – 10 Ověření identity a umožní uživateli přístup do ZAP. |
| Alternativní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Spustí jinou napojenou aplikaci, která ověří jeho identitu a poskytne mu přístup. Při práci s touto aplikací bude chtít přejít do aplikace ZAP, což může udělat kliknutím na příslušné tlačítko. |
| | 2. | Aplikace ZAP | Nebude vyžadovat opětovné přihlášení, využije případ ověření identity z PU – 10 a umožní uživateli přístup. |
| Chybový scénář 1: | Číslo | Aktér | Popis |

| | | | |
|-------------------|-------|----------|---|
| | 1. | Uživatel | Zadá správně přihlašovací údaje, případně se pokusí o přístup do aplikace přes napojený systém. |
| | 2. | Aplikace | Vrací chybovou hlášku „Nedostatečné oprávnění pro přístup!“, jelikož uživatel nemá dostatečná oprávnění pro přístup a navrací uživatele zpět na předchozí stránku. |
| Chybový scénář 2: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Spustí aplikaci a zadá uživatelské jméno a heslo do příslušných polí aplikace a potvrdí tlačítkem. |
| | 2. | Aplikace | Pokus o ověření, však navrací chybovou hlášku „Špatně zadané přihlašovací údaje!“, jelikož údaje byly špatně zadány a navrací uživatele zpět na přihlašovací stránku. |
| Chybový scénář 3: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Spustí aplikaci a zadá uživatelské jméno a heslo do příslušných polí aplikace a potvrdí tlačítkem. |
| | 2. | Aplikace | Pokus o ověření, avšak server je aktuálně nedostupný a navrací chybovou hlášku „Server je aktuálně nedostupný!“ a navrací uživatele zpět na přihlašovací stránku. |

Tab. 5: Příklad užití pro vyhledání v aplikaci ZAP (vlastní zpracování)

| | |
|------------------|---|
| Název: | PU - 2 Vyhledání - pracoviště, žádanky, pacienta |
| Popis: | Uživatel chce při práci s ZAP vyhledat konkrétní žádanku, pacienta či pracoviště. |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Uživatel ZAP (lékař, zdravotní sestra) • Oprávněný uživatel • Uživatel napojeného systému (pracovník laboratoře atp.) |
| Vedlejší aktéři: | |

| | | | |
|-------------------------|---|----------|---|
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému • PU – 10 Ověření identity | | |
| Navazující PU: | | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Zadá do příslušného vyhledávacího pole v ZAP specifické informace např.: OSČ pacienta a potvrdí vyhledávacím tlačítkem, případně překlikne na kartu „Pracoviště“, kde zadá kód pracoviště, pro zobrazení žádanek na pracoviště. |
| | 2. | Aplikace | Již při zadávání „našeptává“ potencionální možnosti, případně pak zobrazuje seznam výsledků, jež odpovídají zadaným kritériím. |
| Alternativní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Vizuálně prohledává seznam všech záznamů v aplikaci. |
| Chybový scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Zadá do vyhledávacího pole chybná data, případně neexistují požadované záznamy. |
| | 2. | Aplikace | Vrací chybovou hlášku „Žádná data k zobrazení!“ a uživateli je umožněn nový pokus o vyhledání požadovaných dat. |

Tab. 6: Příklad užití pro editaci žádanky (vlastní zpracování)

| | |
|------------------|---|
| Název: | PU - 3 Editace žádanky |
| Popis: | Uživatel ZAP chce upravit data na vytvořené žádance. |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Uživatel ZAP (lékař, zdravotní sestra) • Oprávněný uživatel • Uživatel napojeného systému (pracovník laboratoře atp.) |
| Vedlejší aktéři: | |

| | | | |
|-------------------------|--|----------|--|
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému • PU - 2 Vyhledání - pracoviště, žádanky, pacienta • PU - 4 Zobrazení žádanky • PU - 5 Zobrazení žádanek pacienta • PU - 6 Zobrazení žádanek na pracoviště • PU – 10 Ověření identity | | |
| Navazující PU: | | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Po splnění požadavků pro vykonání zadá, změní požadovaná data v žádance a klikne na tlačítko uložit. |
| | 2. | Aplikace | Zpracuje zadaná data a vrátí uživateli hlášku: „Editace proběhla úspěšně, žádanka odeslána ke schválení!“ a vrátí ho zpět na seznam žádanek. |
| Chybový scénář 1: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Klikne na žádanku, jež chce editovat, avšak jiný uživatel portálu ji právě edituje. |
| | 2. | Aplikace | Zobrazí hlášku: „Žádanku edituje OSČ uživatele“ a nepovolí přístup pro editaci. |
| Chybový scénář 2: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Provede všechny kroky hlavního scénáře, avšak při stisknutí tlačítka uložit dojde k chybě na serveru. |
| | 2. | Aplikace | Zobrazí chybovou hlášku: „Data nyní nelze uložit!“ a ponechá uživatele v editačním módu žádanky. |
| Chybový scénář 3: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Provede všechny kroky hlavního scénáře, avšak při vyplňování polí zadá špatná data (např.: špatný formát OSČ) a stiskne tlačítko uložit. |

| | | | |
|--|----|----------|--|
| | 2. | Aplikace | Zobrazí uživateli chybovou hlášku: „Špatný formát dat!“ a neprovede uložení. Uživatele ponechá v editačním módu žádanky. |
|--|----|----------|--|

Tab. 7: Příklad užití pro zobrazení žádanky (Vlastní zpracování)

| | | | |
|-------------------------|--|----------|---|
| Název: | PU - 4 Zobrazení žádanky | | |
| Popis: | Uživatel ZAP chce zobrazit žádanku. Lze zobrazit detail každé jednotlivé žádanky, které jsou filtrovány podle konkrétního pracoviště či pacienta, na nějž byly vystaveny, jak je uvedeno v PU – 5 a PU - 6. Při spuštění jsou nejprve zobrazovány žádanky na pacienta. | | |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Uživatel ZAP (lékař, zdravotní sestra) • Oprávněný uživatel • Uživatel napojeného systému (pracovník laboratoře atp.) | | |
| Vedlejší aktéři: | | | |
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému • PU - 2 Vyhledání - pracoviště, žádanky, pacienta • PU – 10 Ověření identity | | |
| Navazující PU: | <ul style="list-style-type: none"> • PU - 5 Zobrazení žádanek pacienta • PU - 6 Zobrazení žádanek na pracoviště | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Po splnění požadavků pro vykonání rozklikne požadovanou žádanku. |
| | 2. | Aplikace | Zobrazí detail požadované žádanky jako v PU – 3. |
| Alternativní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Klikne na žádanku, jež chce zobrazit, avšak jiný uživatel portálu ji právě edituje. |
| | 2. | Aplikace | Zobrazí hlášku: „Žádanku edituje OSČ uživatele“ a zobrazí detail žádanky pouze pro čtení. |
| Chybový scénář: | Číslo | Aktér | Popis |

| | | | |
|--|----|----------|--|
| | 1. | Uživatel | Provede všechny kroky z hlavního scénáře, avšak při kliknutí pro detail žádanky dojde k chybě na serveru. |
| | 2. | Aplikace | Zobrazí chybovou hlášku: „Data nelze zobrazit!“ a neprovede žádnou akci, uživatele ponechá na místě, kde se v portálu nachází. |

Tab. 8: Příklad užití pro zobrazení žádanek s parametrem (Vlastní zpracování)

| | | | |
|-------------------------|---|----------|---|
| Název: | PU - 5 Zobrazení žádanek pacienta / PU - 6 Zobrazení žádanek na pracoviště | | |
| Popis: | Uživatel ZAP chce zobrazit seznam žádanek pro konkrétního pacienta, či pracoviště. | | |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Uživatel ZAP (lékař, zdravotní sestra) • Oprávněný uživatel • Uživatel napojeného systému (pracovník laboratoře atp.) | | |
| Vedlejší aktéři: | | | |
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU - 4 Zobrazení žádanky | | |
| Navazující PU: | | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Po splnění požadavků pro vykonání rozklikne kartu pracoviště či pacienta. |
| | 2. | Aplikace | Zobrazí detail požadovaného záznamu spolu se seznamem žádanek vytvořených pro dané pracoviště či pacienta. |
| | 3. | Uživatel | Může rozkliknout jednotlivé žádanky z nabízeného seznamu a provádět s nimi další akce. |
| Alternativní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Nevyhledá konkrétního uživatele či pracoviště pomocí PU – 2, ale pomocí filtru zaklikne požadovaná specifika. |

| | | | |
|-----------------|-------|----------|--|
| | 2. | Aplikace | Zobrazí seznamy žádank odpovídající zadaným kritériím ve filtru. |
| | 3. | Uživatel | Může rozkliknout jednotlivé žádanky z nabízeného seznamu a provádět s nimi další akce. |
| Chybový scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Provede všechny kroky z hlavního scénáře, avšak při kliknutí pro detail žádanky dojde k chybě na serveru. |
| | 2. | Aplikace | Zobrazí chybovou hlášku: „Data nelze zobrazit!“ a neprovede žádnou akci, uživatele ponechá na místě, kde se v portálu nachází. |

Tab. 9: Příklad užití pro kopii žádanky (Vlastní zpracování)

| | | | |
|-------------------------|--|-------|-------|
| Název: | PU - 7 Kopie žádanky | | |
| Popis: | Uživatel ZAP chce vytvořit kopii určité žádanky, případně ji upravit a uložit s drobnými změnami. | | |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Uživatel ZAP (lékař, zdravotní sestra) • Oprávněný uživatel • Uživatel napojeného systému (pracovník laboratoře atp.) | | |
| Vedlejší aktéři: | | | |
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému • PU - 2 Vyhledání - pracoviště, žádanky, pacienta • PU - 4 Zobrazení žádanky • PU - 5 Zobrazení žádanek pacienta • PU - 6 Zobrazení žádanek na pracoviště • PU – 10 Ověření identity | | |
| Navazující PU: | | | |
| Hlavní scénář: | Číslo | Aktér | Popis |

| | | | |
|----------------------|-------|----------|--|
| | 1. | Uživatel | Po splnění požadavků pro vykonání klikne tlačítko pro duplikaci (kopii) žádanky. |
| | 2. | Aplikace | Zobrazí v novém okně předvyplněná pole duplikované žádanky, podle zvoleného typu. |
| | 3. | Uživatel | Upraví, případně vyplní další požadovaná pole a stiskne tlačítko pro uložení. |
| | 4. | Aplikace | Zobrazí hlášku: „Žádanka úspěšně zkopírována a uložena.“ |
| | 5. | Uživatel | Je po uložení navrácen na původní žádanku a může dále pokračovat v práci s ZAP. |
| Alternativní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Po splnění požadavků pro vykonání klikne na tlačítko pro duplikaci (kopii) žádanky rovnou ve zobrazeném seznamu žádanek a zvolí, v jakém z nabízených typů se má žádanka duplikovat. |
| | 4. | Aplikace | Zobrazí v novém okně předvyplněná pole duplikované žádanky, podle zvoleného typu. |
| | 5. | Uživatel | Upraví, případně vyplní další požadovaná pole a stiskne tlačítko pro uložení. |
| | 6. | Aplikace | Zobrazí hlášku: „Žádanka úspěšně zkopírována a uložena.“ |
| | 7. | Uživatel | Je po uložení navrácen na původní žádanku a může dále pokračovat v práci s ZAP. |
| Chybový scénář 1: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Po splnění požadavků pro vykonání klikne na tlačítko pro duplikaci (kopii) žádanky rovnou ve zobrazeném seznamu žádanek, avšak žádanku nyní edituje některý z uživatelů ZAP. |
| | 2. | Aplikace | Zobrazí hlášku: „Žádanka nelze zkopírovat, je právě editována uživatelem OSČ!“ |
| Chybový scénář 2: | Číslo | Aktér | Popis |

| | | | |
|-------------------|-------|----------|---|
| | 1. | Uživatel | Provede všechny kroky hlavního či alternativního scénáře, avšak při vyplňování dalších dat zadá chybná data (např.: špatný formát OSČ) a stiskne tlačítko uložit. |
| | 2. | Aplikace | Zobrazí uživateli chybovou hlášku: „Špatný formát dat!“ a neprovede uložení. Uživatele ponechá v novém okně duplikované žádanky. |
| Chybový scénář 3: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Provede všechny kroky hlavního či alternativního scénáře, avšak při stisknutí tlačítka uložit dojde k chybě na serveru. |
| | 2. | Aplikace | Zobrazí chybovou hlášku: „Data nyní nelze uložit!“ a uživatele ponechá v novém okně duplikované žádanky. |

Tab. 10: Příklad užití pro smazání žádanky (Vlastní zpracování)

| | |
|-------------------------|--|
| Název: | PU - 8 Smazání žádanky |
| Popis: | Uživatel ZAP chce smazat určitou žádanku. |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Uživatel ZAP (lékař, zdravotní sestra) • Oprávněný uživatel • Uživatel napojeného systému (pracovník laboratoře atp.) |
| Vedlejší aktéři: | |
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému • PU - 2 Vyhledání - pracoviště, žádanky, pacienta • PU - 4 Zobrazení žádanky • PU - 5 Zobrazení žádanek pacienta • PU - 6 Zobrazení žádanek na pracoviště • PU – 10 Ověření identity |
| Navazující PU: | |

| | | | |
|----------------------|-------|----------|--|
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Po splnění požadavků pro vykonání zaklikne tlačítko pro smazání žádanky. |
| | 2. | Aplikace | Zobrazí hlášku: „Žádanka úspěšně smazána.“ |
| Alternativní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Po splnění požadavků pro vykonání klikne na tlačítko pro smazání žádanky rovnou ve zobrazeném seznamu žádanek. |
| | 2. | Aplikace | Zobrazí hlášku: „Žádanka úspěšně smazána.“ |
| Chybový scénář 1: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Po splnění požadavků pro vykonání klikne na tlačítko pro smazání žádanky rovnou ve zobrazeném seznamu žádanek, avšak žádanku nyní edituje některý z uživatelů ZAP. |
| | 2. | Aplikace | Zobrazí hlášku: „Žádanka nelze smazat, je právě editována uživatelem OSČ!“ |
| Chybový scénář 2: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Provede všechny kroky hlavního scénáře, avšak při stisknutí tlačítka smazat žádanku dojde k chybě na serveru. |
| | 2. | Aplikace | Zobrazí chybovou hlášku: „Data nyní nelze smazat!“ a ponechá uživatele ve stavu při kterém kliknul na tlačítko pro odstranění žádanky. |

Tab. 11: Příklad užití pro založení nové žádanky (Vlastní zpracování)

| | |
|----------------|---|
| Název: | PU - 9 Založení nové žádanky |
| Popis: | Uživatel ZAP chce založit novou žádanku. |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Uživatel ZAP (lékař, zdravotní sestra) • Oprávněný uživatel • Uživatel napojeného systému (pracovník laboratoře atp.) |

| | | | |
|-------------------------|---|----------|--|
| Vedlejší aktéři: | | | |
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému • PU - 2 Vyhledání - pracoviště, žádanky, pacienta • PU – 10 Ověření identity | | |
| Navazující PU: | | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Po splnění požadavků pro vykonání klikne na tlačítko pro založení nové žádanky rovnou v seznamu zobrazených žádanek a zvolí jednu z nabízených typů žádanek. |
| | 2. | Aplikace | Zobrazí prázdná pole zvoleného typu žádanky. |
| | 3. | Uživatel | Vyplní potřebná pole a stiskne tlačítko pro uložení. |
| | 4. | Aplikace | Zobrazí hlášku: „Žádanka úspěšně uložena a odeslána k potvrzení.“ |
| Chybový scénář 1: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Provede všechny kroky hlavního scénáře, avšak při vyplňování dat zadá chybná data (např.: špatný formát OSČ) a stiskne tlačítko uložit. |
| | 2. | Aplikace | Zobrazí uživateli chybovou hlášku: „Špatný formát dat!“ a neprovede uložení. Uživatele ponechá ve stavu zakládání nové žádanky. |
| Chybový scénář 2: | Číslo | Aktér | Popis |
| | 1. | Uživatel | Provede všechny kroky hlavního scénáře, avšak při stisknutí tlačítka uložit dojde k chybě na serveru. |
| | 2. | Aplikace | Zobrazí chybovou hlášku: „Data nyní nelze uložit!“ a ponechá uživatele ve stavu zakládání nové žádanky. |

Tab. 12: Příklad užití pro ověření identity (Vlastní zpracování)

| | | | |
|-------------------------|--|----------|--|
| Název: | PU – 10 Ověření identity | | |
| Popis: | Proces následující po zadání přihlašovacích údajů uživatele a jejich potvrzení tlačítkem. | | |
| Hlavní aktéři: | <ul style="list-style-type: none"> • IDM | | |
| Vedlejší aktéři: | <ul style="list-style-type: none"> • Správce ZAP • Uživatel ZAP (lékař, zdravotní sestra) • Oprávněný uživatel • Uživatel napojeného systému (pracovník laboratoře atp.) | | |
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému | | |
| Navazující PU: | | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Aplikace | Po přijetí přihlašovacích údajů uživatele, jak je uvedeno v případu užití pro Přihlášení uživatele, ověřuje jejich správnost vůči IDM. |
| | 2. | IDM | Ověří zaslané informace a v případě shody zasílá nazpět oprávnění přiřazená pro daného uživatele. |
| | 3. | Aplikace | Přijímá data o uživateli a na základě nich zpřístupňuje další části ZAP, jak je uvedeno v PU - 1. |
| Chybový scénář: | Číslo | Aktér | Popis |
| | 1. | Aplikace | Po přijetí přihlašovacích údajů uživatele, jak je uvedeno v případu užití pro Přihlášení uživatele, ověřuje jejich správnost vůči IDM. |
| | 2. | IDM | Se pokouší o spojení se serverem a ověření údajů, avšak server je aktuálně nedostupný. Tudíž nazpět zasílá chybovou hlášku: „Server je současně nedostupný!“ |
| | 3. | Aplikace | Přijímá zaslou hlášku serverem a zobrazuje ji uživateli. |

Tab. 13: Příklad užití pro tisk žádanky (Vlastní zpracování)

| | | | |
|-------------------------|--|--------------------|---|
| Název: | PU - 11 Tisk žádanky | | |
| Popis: | Oprávněný uživatel ZAP chce vytisknout určitou žádanku. | | |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Oprávněný uživatel | | |
| Vedlejší aktéři: | | | |
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému • PU - 2 Vyhledání - pracoviště, žádanky, pacienta • PU - 4 Zobrazení žádanky • PU - 5 Zobrazení žádanek pacienta • PU - 6 Zobrazení žádanek na pracoviště • PU – 10 Ověření identity | | |
| Navazující PU: | | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Oprávněný uživatel | Po splnění požadavků pro vykonání zaklikne tlačítko pro tisk žádanky, jenž je dostupné pouze v oprávněném módu a zvolí šablonu v níž chce žádanku vytisknout. |
| | 2. | Aplikace | Zobrazí hlášku: „Žádanka úspěšně odeslána na tisk.“ |
| Alternativní scénář: | Číslo | Aktér | Popis |
| | 1. | Oprávněný uživatel | Po splnění požadavků pro vykonání klikne na tlačítko pro tisk žádanky, které se zpřístupní pouze v oprávněném módu, rovnou ve zobrazeném seznamu žádanek. |
| | 2. | Aplikace | Zobrazí hlášku: „Žádanka úspěšně odeslána na tisk.“ |
| Chybový scénář 1: | Číslo | Aktér | Popis |

| | | | |
|-------------------|-------|--------------------|---|
| | 1. | Oprávněný uživatel | Po splnění požadavků pro vykonání klikne na tlačítko pro tisk žádanky rovnou ve zobrazeném seznamu žádanek, avšak žádanku nyní edituje některý z uživatelů ZAP. |
| | 2. | Aplikace | Zobrazí hlášku: „Žádanka nelze vytisknout, je právě editována uživatelem OSC!“ a ponechá uživatele ve stavu, při kterém kliknul na tlačítko pro tisk žádanky. |
| Chybový scénář 2: | Číslo | Aktér | Popis |
| | 1. | Oprávněný uživatel | Provede všechny kroky hlavního scénáře, avšak při stisknutí tlačítka vytisknout žádanku dojde k chybě na serveru. |
| | 2. | Aplikace | Zobrazí chybovou hlášku: „Data nyní nelze tisknout!“ a ponechá uživatele ve stavu, při kterém kliknul na tlačítko pro tisk žádanky. |

Tab. 14: Příklad užití pro odeslání žádanky (Vlastní zpracování)

| | |
|-------------------------|--|
| Název: | PU - 12 Odeslání žádanky |
| Popis: | Oprávněný uživatel ZAP schválí a odešle vytvořené žádanky od běžných uživatelů ZAP k vyřízení. |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Oprávněný uživatel |
| Vedlejší aktéři: | |
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému • PU - 2 Vyhledání - pracoviště, žádanky, pacienta • PU - 4 Zobrazení žádanky • PU - 5 Zobrazení žádanek pacienta • PU - 6 Zobrazení žádanek na pracoviště • PU – 10 Ověření identity |

| | | | |
|------------------------|-------|--------------------|---|
| Navazující PU: | | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Oprávněný uživatel | Po splnění požadavků pro vykonání klikne tlačítko pro zobrazení žadanek ke schválení/odeslání. |
| | 2. | Aplikace | Zobrazí seznam žadanek, které jsou uloženy a připravené ke schválení. |
| | 3. | Oprávněný uživatel | Označí všechny žádanky, jenž chce potvrdit a odeslat a klikne na tlačítko pro odeslání. |
| | 4. | Aplikace | Zobrazí hlášku: „Odeslání proběhlo úspěšně!“ a odeslané žádanky označí jako odeslané. Uživatele pak ponechá na místě, odkud kliknul na tlačítko pro odeslání. |
| Alternativní scénář 1: | Číslo | Aktér | Popis |
| | 1. | Oprávněný uživatel | Po splnění požadavků pro vykonání klikne tlačítko pro odeslání žádanky, jenž je dostupné pouze v oprávněném módu. |
| | 2. | Aplikace | Zobrazí hlášku: „Žádanka úspěšně odeslána.“ |
| Alternativní scénář 2: | Číslo | Aktér | Popis |
| | 1. | Oprávněný uživatel | Po splnění požadavků pro vykonání klikne na tlačítko pro odeslání žádanky, které se zpřístupní pouze v oprávněném módu, rovnou ve zobrazeném seznamu žadanek. |
| | 2. | Aplikace | Zobrazí hlášku: „Žádanka úspěšně odeslána.“ |
| Chybový scénář 1: | Číslo | Aktér | Popis |
| | 1. | Oprávněný uživatel | Po splnění požadavků pro vykonání klikne na tlačítko pro odeslání žádanky rovnou ve zobrazeném seznamu žadanek, avšak žádanku nyní edituje některý z uživatelů ZAP. |

| | | | |
|-------------------|-------|--------------------|---|
| | 2. | Aplikace | Zobrazí hlášku: „Žádanku nelze odeslat, je právě editována uživatelem OSČ!“ a ponechá uživatele ve stavu při kterém kliknul na tlačítko pro tisk žádanky. |
| Chybový scénář 2: | Číslo | Aktér | Popis |
| | 1. | Oprávněný uživatel | Provede všechny kroky hlavního scénáře, avšak při stisknutí tlačítka odeslat žádanku dojde k chybě na serveru. |
| | 2. | Aplikace | Zobrazí chybovou hlášku: „Data nyní nelze odeslat!“ a ponechá uživatele ve stavu při kterém kliknul na tlačítko pro odeslání žádanky. |

Tab. 15: Příklad užití pro tvorbu nových typů žádenek (Vlastní zpracování)

| | | | |
|-------------------------|---|-------------|---|
| Název: | PU - 13 Tvorba nových typů žádenek | | |
| Popis: | Správce ZAP chce vytvořit nový typ žádanky s novými parametry. | | |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Správce ZAP | | |
| Vedlejší aktéři: | | | |
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému • PU – 10 Ověření identity | | |
| Navazující PU: | | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Správce ZAP | Po splnění požadavků pro vykonání klikne na tlačítko pro vytvoření nového typu žádanky, které je dostupné pouze v administrátorském rozhraní. |
| | 2. | Aplikace | Zobrazí potřebná pole a funkcionality pro tvorbu nového typu žádanky. |
| | 3. | Správce ZAP | Vyplní potřebná data a pomocí kódu vydefinuje vzhled a parametry nového typu žádanky a stiskne tlačítko pro uložení. |

| | | | |
|-----------------|-------|-------------|--|
| | 4. | Aplikace | Zobrazí hlášku: „Nový typ žádanky byl úspěšně vytvořen!“ Uživatele pak ponechá na místě, odkud kliknul na tlačítko pro uložení. |
| Chybový scénář: | Číslo | Aktér | Popis |
| | 1. | Správce ZAP | Provede všechny kroky hlavního scénáře, avšak při stisknutí tlačítka pro uložení nového typu žádanky dojde k chybě na serveru. |
| | 2. | Aplikace | Zobrazí chybovou hlášku: „Žádanku nyní nelze uložit!“ a ponechá uživatele ve stavu při kterém kliknul na tlačítko pro uložení nového typu žádanky. |

Tab. 16: Příklad užití pro tvorbu nových tiskových šablon (Vlastní zpracování)

| | | | |
|-------------------------|---|-------------|--|
| Název: | PU - 14 Tvorba nových tiskových šablon | | |
| Popis: | Správce ZAP chce vytvořit novou šablonu pro tisk žádanky. | | |
| Hlavní aktéři: | <ul style="list-style-type: none"> • Správce ZAP | | |
| Vedlejší aktéři: | | | |
| Požadavky pro vykonání: | <ul style="list-style-type: none"> • PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému • PU – 10 Ověření identity | | |
| Navazující PU: | | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Správce ZAP | Po splnění požadavků pro vykonání klikne na tlačítko pro vytvoření nové tiskové šablony, které je dostupné pouze v administrátorském rozhraní. |
| | 2. | Aplikace | Zobrazí potřebná pole a funkcionality pro tvorbu nové tiskové šablony. |
| | 3. | Správce ZAP | Nadefinuje vzhled a parametry nové šablony a stiskne tlačítko pro uložení. |

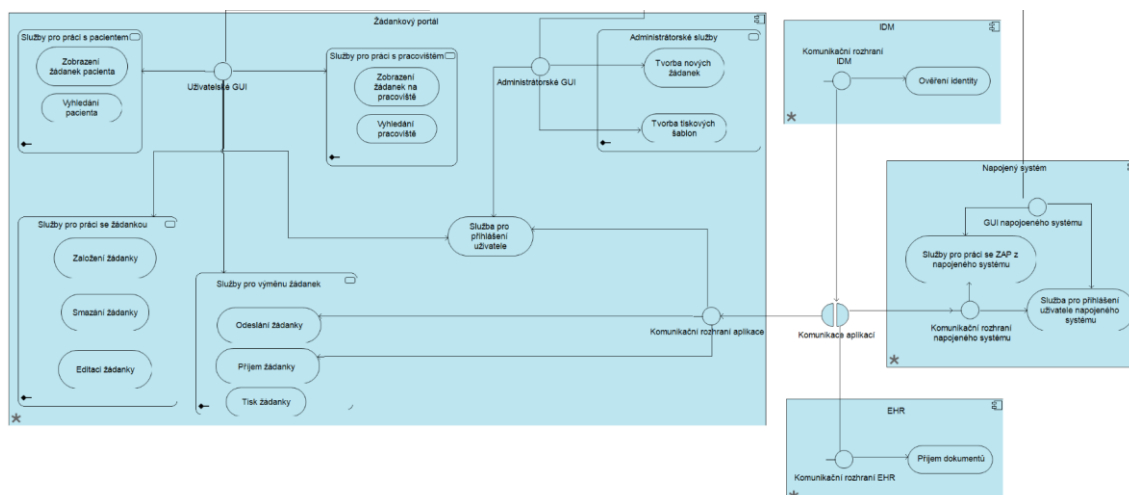
| | | | |
|-----------------|-------|-------------|--|
| | 4. | Aplikace | Zobrazí hlášku: „Nová šablona byla úspěšně vytvořena!“ Uživatelé pak ponechá na místě, odkud kliknul na tlačítko pro uložení. |
| Chybový scénář: | Číslo | Aktér | Popis |
| | 1. | Správce ZAP | Provede všechny kroky hlavního scénáře, avšak při stisknutí tlačítka pro uložení nové šablony dojde k chybě na serveru. |
| | 2. | Aplikace | Zobrazí chybovou hlášku: „Tiskovou šablonu nyní nelze uložit!“ a ponechá uživatele ve stavu, při kterém kliknul na tlačítko pro uložení šablony. |

Tab. 17: Příklad užití pro práci s aplikací ZAP z napojeného IS (Vlastní zpracování)

| | | | |
|-------------------------|---|-----------------------------|---|
| Název: | PU - 15 Práce s napojeným systémem | | |
| Popis: | Uživatel napojeného systému chce při práci s ním využít i služby či data ZAP. | | |
| Hlavní aktéři: | <ul style="list-style-type: none"> Uživatel napojeného systému (pracovník laboratoře atp.) | | |
| Vedlejší aktéři: | | | |
| Požadavky pro vykonání: | <ul style="list-style-type: none"> PU – 1 Přihlášení k ZAP / PU – 16 Přihlášení k napojenému systému PU – 10 Ověření identity | | |
| Navazující PU: | | | |
| Hlavní scénář: | Číslo | Aktér | Popis |
| | 1. | Uživatel napojeného systému | Po splnění požadavků pro vykonání započne uživatel práci s napojeným systémem při této práci chce využít ZAP, stejným způsobem a se stejnými službami jako klasický uživatel ZAP. |
| | 2. | Aplikace | Poskytuje potřebné prostředí pro plnění těchto služeb, stejně jako tomu bylo v předešlých případech užití. |

3.3 Aplikační vrstva

V této kapitole bude popsána aplikační vrstva celého návrhu spolu s fungováním aplikace na pozadí, pro vysvětlení a pochopení všech závislostí a funkcí budou uvedeny i konkrétní příklady obsluhy určitých částí aplikace či služeb, jenž byly navrženy v business vrstvě aplikace.



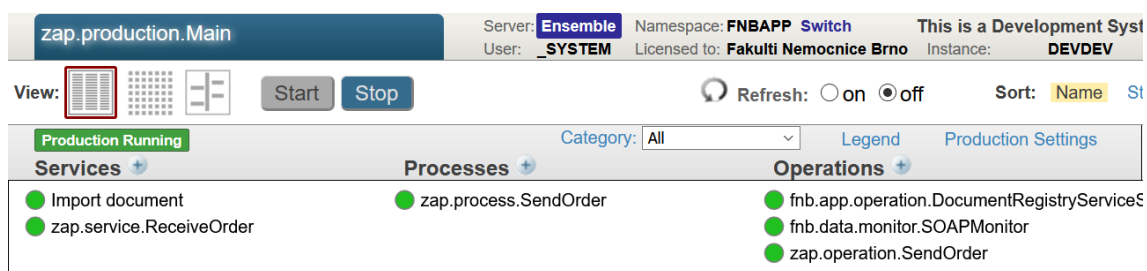
Obrázek č. 6: Diagram pro aplikační vrstvu aplikace (Vlastní zpracování)

3.3.1 Realizace případů užití

Zde budou uvedena řešení jednotlivých částí aplikace od nejvyšších souhrnných funkcí až po různé specifické služby obsluhující procesy uvedené v business vrstvě a jednotlivých případech užití, vše by mělo vést k funkčním návrhům, které by měly zajistit běh celé aplikace a objasnění jejího chodu.

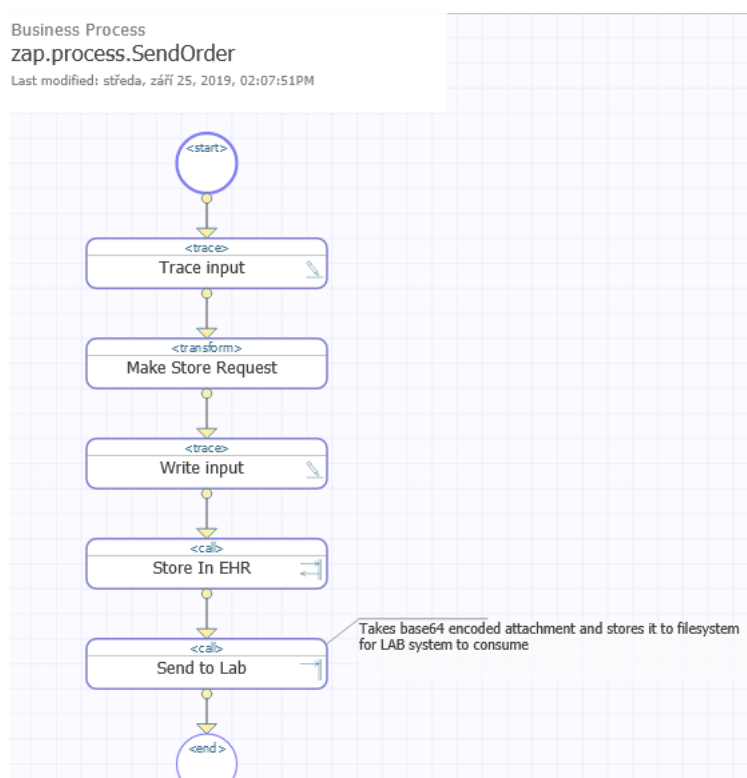
3.3.1.1 Nastavení komponent produkce pro aplikaci

Pro správnou interakci celé aplikace s okolím je velmi důležité také nastavení a propojení jednotlivých komponent uvnitř produkce pro komunikaci s dalšími aplikacemi, databázemi atp. Aktuální propojení je nastaveno následujícím způsobem:



Obrázek č. 7: Nastavení a komponenty produkce (Vlastní zpracování)

Obě služby přijímají data, jedna jako SOAPová služba, druhá pomocí souborů. Data jsou pak zpracovávána v procesu **SendOrder**, jehož činnost vypadá následovně:



Obrázek č. 8: Proces SendOrder (Vlastní zpracování)

Následně jsou data předávána do operací, což je vidět i v diagramu zmiňovaného procesu, které tato data např.: transformují do standardizovaného xml dokumentu a uloží do EHR (Electronic Health Record), případně přepošlou do jiných aplikací čekajících na daná data. Kvůli tomuto předávání dat obsahuje ZAP také služby pro příjem, či odesílání žádank, pro případnou obsluhu dalšího napojeného systému, který by žádanky jak vytvářel, tak přijímal.

3.3.1.2 Princip volání služeb v aplikaci

Jedním z hlavních bodů je rozhraní pro volání webových služeb. Pro jeho vytvoření a nastavení je využíváno nástroje z firmy InterSystems, kterým je Management portál, v něm si vytvoříme novou Web aplikaci, u které provedeme potřebná nastavení. Ta mohou vypadat pro náš účel následovně:

Edit definition for web application /RestApi/zap:

General Application Roles Matching Roles

Name: /RestApi/zap
Required (e.g. /csp/appname)

Description: Zadankovy portal

Namespace: FNBAPP Default Application for FNBAPP: /csp/fnbapp ☐ Namespace Default Application

Enabled: ☒ Application ☒ CSP/ZEN ☒ Inbound Web Services
☐ DeepSee ☐ iKnow

Permitted Classes:

Security Settings: Resource Required: Group By ID: Allowed Authentication Methods: ☐ Unauthenticated ☒ Password ☐ Kerberos ☒ Delegated ☒ LDAP ☐ Login Cookie

Session Settings: Session Timeout: 900 seconds Event Class: Use Cookie for Session: Always Session Cookie Path: /RestApi/zap/

Dispatch Class: zap.rest.Router

CSP File Settings: Serve Files: Always Serve Files Timeout: 3600 seconds CSP Files Physical Path: Package Name: Default Superclass: CSP Settings: ☒ Recurse ☒ Auto Compile ☒ Lock CSP Name

Custom Pages: Login Page: Change Password Page: Custom Error Page:

Obrázek č. 9: Nastavení rozhraní v Management portálu (Vlastní zpracování)

V poli „Dispatch class“ můžeme vidět uvedenou třídu, na kterou se toto rozhraní odkazuje. V této třídě je pak napsán kód pro jednotlivé RESTové služby. V samotné třídě pak funguje přesměrování pomocí značky „<Route />“, kde v parametrech tohoto tagu jsou uvedeny jednotlivé parametry, jejichž význam je následující:

Tab. 18: Tabulka parametrů tagu Route (Vlastní zpracování)

| Název | Popis |
|--------|---|
| URL | Specifikuje formát poslední části URL adresy pro vyvolání určité RESTové služby. |
| Method | Určuje metodu požadavku HTTP pro volání RESTu, obvykle se jedná o GET, POST, PUT nebo DELETE, ale lze použít libovolnou |

| | |
|------|--|
| | metodu požadavku HTTP, která by měla být zvolena s ohledem na vhodnost námi vytvořené metody volané službou. |
| Call | Určuje metodu třídy, která se má volat RESTovou službou. |
| Cors | Tento parametr je volitelný a nabývá hodnot „true“ nebo „false“ a řídí zpracování záhlaví CORS |

Tyto parametry pak lépe definují a zajišťují tak správnou funkci tohoto „směrovače“ uvnitř třídy tak, že vždy zavolá odpovídající kód, který je třeba pro vykonání dané činnosti. Na následujícím obrázku můžeme vidět část sekce se „směrovači“, které odkazují ke konkrétním metodám v této třídě:

```

<Routes>
  <!-- Zadanky -->
  <Route Url="/orders" Method="GET" Call="ZadankySeznam" Cors="true" />
  <Route Url="/orders/batch" Method="POST" Call="ZadankySend" Cors="true" />
  <Route Url="/ZadankyTisk" Method="POST" Call="ZadankyTisk" Cors="true" />
  <!-- Vybrana zadanka -->
  <Route Url="/orders/form/type" Method="GET" Call="ZadankaNovaTyp" Cors="true" />
  <Route Url="/ZadankaNovaForm" Method="POST" Call="ZadankaNovaForm" Cors="true" />
  <Route Url="/orders/state" Method="POST" Call="ZadankaStav" Cors="true" />
  <Route Url="/ZadankaStorno" Method="POST" Call="ZadankaStorno" Cors="true" />
  <Route Url="/ZadankaTisk" Method="POST" Call="ZadankaTisk" Cors="true" />
  <Route Url="/ZadankaSend" Method="POST" Call="ZadankaSend" Cors="true" />
</Routes>

```

Obrázek č. 10: Kód routeru pro volání jednotlivých metod (Vlastní zpracování)

Jedna z metod, která je volána při přechodu na jedno konkrétní URL, jenž je uvedeno výše, je v detailu taktéž zobrazena na přiloženém obrázku. Jedná se o metodu pro vytvoření nové žádanky a je napsána v programovacím jazyce Caché ObjectScript. Stejným způsobem jsou pak vyřešeny všechny potřebné služby zmiňované v návrhu aplikace ZAP.

```

/// Zadanka - nova, zadani podle formulare/sablony
ClassMethod ZadankaNovaForm() As %Status
{
    n %zFLP,%zIDX,%zDta
    s GbR=..TempGlobal()
    s GbI=$na(@GbR@("I"))
    s ID=$g(@GbI@("id"))
    s IDA=$g(@GbI@("ida"))
    ;Provizorne - ulozeni dat. Novy -> IDX=""
    s %zIDX=""
    s Err=$$Load^appzapZadanka(ID,IDA,%zIDX)
    ;Naplneni dat
    d TRANSFORM^LIBdatop(GbI,"%zFLP","cislo:CisloZadanky,ordinoval:Ordinoval,poznamka:Poznamka,dg:DG")
    i $g(@GbI@("priorita")) s %zFLP("Statim")=1
    e s %zFLP("Statim")=""
    ;Pripadne doplneni, prepocty,...
    ;Ulozeni
    d ..TempGarbage(GbR)
    i 'Err s Err=$$Save^appzapZadanka(ID,IDA,%zIDX)
    i Err $$$RestStatus500("Chyba " _Err,"Neco je spatne...")
    s %zDta("id")=ID,%zDta("ida")=IDA,%zDta("idz")=%zIDX
    s DataO=$$Encode^LIBjson("%zDta")
    $$$RestStatus200String(DataO)
}

```

Obrázek č. 11: Kód metody pro službu vytvoření nové žádanky (Vlastní zpracování)

Toto fungování aplikace na pozadí je pak pomocí javascriptového webového frameworku Angular propojeno pro interakci s uživatelem, s uživatelským rozhraním aplikace. Napojení a přesměrování na metody a kód uložený v Caché na serveru je následovné; pomocí vytvořeného souboru pro nastavení adres do „proměnných“.

```

1 {
2   "/Restapi/zap": {
3     "target": "http://192.168.62.4:9080/",
4     "secure": false,
5     "changeOrigin": true,
6     "pathRewrite": {
7       "^/api/auth": ""
8     }
9   },
10  "/api/sigan": {
11    "target": "https://192.168.62.4:10443/api/sigan/",
12    "secure": false,
13    "changeOrigin": true,
14    "pathRewrite": {
15      "^/api/sigan": ""
16    }
17  },
18  "/api/..."

```

Obrázek č. 12: Soubor pro nastavení adres v Angularu (Vlastní zpracování)

V Angularu se pak k adresám přistupuje pomocí námi vytvořeného zápisu, který se pak překládá na skutečnou adresu na server, kde si dané volání přebírá již zmiňovaná webová aplikace s odkazem na konkrétní třídu.

3.3.1.3 Tvorba formulářů

Podstatnou částí celého systému jsou také již zmíněné formuláře, které se využívají při vyplňování všech žádank poskytovanych na portálu. Tvorbu formulářů bude v při běhu portálu obstarávat jeho správce, formuláře se pak dají definovat v Caché pomocí xml tagů a jejich parametrů, podobně jako v html, volitelnost komponent a vzhledu formulářů je téměř neomezená, tudíž záleží především na potřebách uživatelů žádankového portálu.

Pro tvorbu nového formuláře je v první řadě nutné vytvořit si novou třídu v Caché. V ní pak definujeme formulář hlavním párovým tagem **<Form>**. Tento tag může obsahovat mnoho další parametrů:

Tab. 19: Parametry tagu <Form> (Vlastní zpracování)

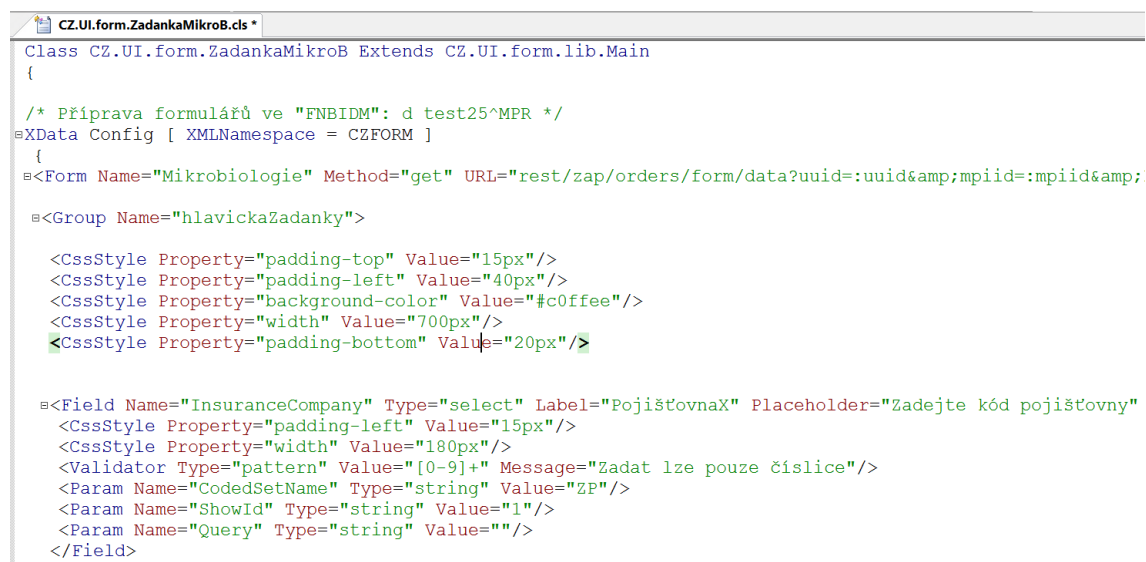
| Název | Popis |
|--------|--|
| Name | Jméno formuláře (žádanky), které se bude zobrazovat. |
| Method | Určuje metodu požadavku HTTP pro volání RESTu, obvykle se jedná o GET nebo POST. |
| URL | Určuje url adresu, odkud bude daný formulář dostupný, v url adrese pak mohou být obsaženy další parametry, jako id atp... |

Uvnitř tohoto hlavního tagu jsou pak další tagy a to především **<Group>** značící skupinu prvků ve formuláři, obvykle s nastavením pojmenování skupiny nebo jejího skrytí, a **<Field>**, který označuje jednotlivé pole formuláře, jehož typ je pak určen parametrem.

Tab. 20: Parametry tagu <Field> (Vlastní zpracování)

| Název | Popis |
|-------------|--|
| Name | Jméno pojmenování prvku ve formuláři. |
| Type | Druh zobrazovaného pole ve formuláři, může být např.: input, checkbox, label atd.. |
| Label | Popisek, který bude u políčka zobrazen. |
| Position | Pozice, na které bude prvek ve formuláři vykreslen. |
| Placeholder | Popisek zobrazovaný uvnitř pole, pokud se jedná o input. |

Uvnitř tagu **<Field>** se pak mohou vyskytovat tagy **<CssStyle>** - pro nastavení css stylu, **<Param>** - pro nastavení dalších parametrů, např.: pro načtení hodnoty z databáze či **<Validator>** - pro ověření správnosti zadaných hodnot. Praktický příklad pak může vypadat následovně:



```

Class CZ.UI.form.ZadankaMikroB Extends CZ.UI.form.lib.Main
{
    /* Příprava formulářů ve "FNBIDM": d test25^MPR */
    XData Config [ XMLNamespace = CZFORM ]
    {
        <Form Name="Mikrobiologie" Method="get" URL="rest/zap/orders/form/data?uuid=:uuid&mpiid=:mpiid&";
        <Group Name="hlavickaZadanky">
            <CssStyle Property="padding-top" Value="15px"/>
            <CssStyle Property="padding-left" Value="40px"/>
            <CssStyle Property="background-color" Value="#c0ffee"/>
            <CssStyle Property="width" Value="700px"/>
            <CssStyle Property="padding-bottom" Value="20px"/>

            <Field Name="InsuranceCompany" Type="select" Label="PojišťovnaX" Placeholder="Zadejte kód pojišťovny"
                <CssStyle Property="padding-left" Value="15px"/>
                <CssStyle Property="width" Value="180px"/>
                <Validator Type="pattern" Value="[0-9]+" Message="Zadat lze pouze číslice"/>
                <Param Name="CodedSetName" Type="string" Value="ZP"/>
                <Param Name="ShowId" Type="string" Value="1"/>
                <Param Name="Query" Type="string" Value="" />
            </Field>
        </Group>
    }
}

```

Obrázek č. 13: Definice formuláře v Caché (Vlastní zpracování)

Nakonec je vložen i tag **<Button>**, který značí tlačítko, obvykle pro uložení formuláře, jehož parametry jsou takřka stejné jako zmiňované parametry u tagu **<Form>**, opět se jedná o metodu odeslání dat, pojmenování atp... Navíc jsou však uvnitř v tomto tagu obsaženy tagy **<FormMap>**, které určují kam se budou data, specifikovaná v tomto tagu, ukládat do databáze.

Tab. 21: Parametry tagu **<FormMap>** (Vlastní zpracování)

| Název | Popis |
|----------------|---|
| OutputProperty | Označuje pojmenování proměnné v databázi, kam se má hodnota uložit. |
| FormProperty | Nabývá hodnot např.: „hlavickaZadanky.PatientName“, což znamená označení prvku ve formuláři, ze kterého se má hodnota vzít. |

Ve formuláři je však nutné nastavit také vkládání dat do polí, která v sobě při zobrazení formuláře mají mít nastavenou určitou, předem definovanou hodnotu, která je získána z databáze. Pro tento účel se využívají tagy **<Map>**, které se vkládají do nadřazeného tagu **<Maps>**. Specifikace pak opět probíhá pomocí parametrů:

Tab. 22: Parametry tagu <Map> (Vlastní zpracování)

| Název | Popis |
|--------------|---|
| Name | Označuje pojmenování prvku ve formuláři pro specifikaci, kam data vložit. |
| ParentName | Značí pojmenování nadřazené skupiny ve formuláři, v níž je umístěn prvek, do kterého se mají hodnoty vkládat. |
| JsonSelector | Označuje konkrétní data ve struktuře JSONu, která se mají do daného prvku vložit. |

3.3.1.4 Tvorba tiskových šablon

Jak již bylo zmíněno v předchozích kapitolách, především tedy v případech užití, další významnou funkcionalitou je tvorba nových tiskových šablon. Nejprve je nutné vydefinovat si požadovanou strukturu nové šablony v XML manuálně, což spadá do činností správce ZAP. Základní struktura šablony je dána především tagem **<Template>** a **<header>**, ostatní prvky šablony jsou plně volitelné a jedná se především o prvky v ohraničení tagů **<const>**, **<transformations>**, **<bands>**. Pro bližší specifikaci využití těchto tagů v šabloně je popíši v rozdělení níže:

<const> - do tohoto párového tagu se umísťují jednotlivé konstanty, které chceme používat uvnitř šablony, příklad pro využití této sekce může být při výskytu více položek na jednom řádku a my potřebujeme tento řádek posunout, bez využití konstanty budeme muset mnohokrát měnit souřadnici Y, i když se její hodnota mění u všech položek stejně. Při využití konstanty pak stačí přepsat její hodnotu v definici konstanty a hodnota se přepíše ve všech odkazech na tuto konstantu. Odkaz na konstantu se pak indikuje znakem **%** a následné přiřazení k souřadnici Y by mohlo vypadat následovně **y = "%const"**, využívat lze i jednu šablonu pro více funkcí, kdy konstantu mění program, jež volá danou šablonu, taková konstanta pak začíná znakem **@**. Tato sekce je nepovinná, tudíž nemusí být v šabloně vůbec použita.

<transformations> - v tomto párovém tagu se nacházejí definice „výpočtových dat“, případně k modifikaci stávajících podle požadavků tvůrce šablony. Opět se jedná o nepovinnou sekci. Transformací může být více, vyvolávají se buď na začátku nebo např. při každém vyvolání tabulkového bandu. Vše závisí na parametrech jednotlivých transformací a prvcích, jež obsahují, obojí je uvedeno v tabulkách níže:

Tab. 23: Parametry tagu <transformations> (Vlastní zpracování)

| Název parametru | Popis |
|-----------------|--|
| Id | Označuje ID jednotlivých transformací. |
| Name | Vlastní pojmenování transformace. |
| Type | Určuje typ transformace: 1 – dojde ke zpracování na začátku tisku šablony, 2 – zpracovává až band, který si transformaci sám vyvolá. |

Tab. 24: Položky tagu <transformations> (Vlastní zpracování)

| Název položky | Popis |
|---------------|---|
| Id | Označuje ID jednotlivých položek. |
| Name | Vlastní pojmenování položky. |
| Type | Určuje typ akce 0 - naplnění, 1 - konkatenace (spojení). |
| Source | Označuje, kam se uloží výsledek celé operace. |
| Op | Operace, určující hodnotu |
| | 0 obsah arg1. |
| | 1 data z pole, určených arg1. |
| | 2 podřetězec z pole určených arg1, arg2 a arg3 určují startovací a konečnou pozici. |
| | 3 piece (část textu podle oddělovače) z pole určených arg1, oddělovač je v arg2 a číslo části v arg3. |

| | | | |
|-----------------|---|---|--|
| | | 4 | konstanta určená arg1, možné hodnoty: 0 - číslo stránky, 2 - celkový počet stránek |
| arg1, ..., arg3 | Označuje argumenty operace, jejich význam na ni záleží. | | |

<**bands**> - jedná se o základní sekci celé šablony, do níž jsou promítána data. Souřadnice bandu se berou od horního levého rohu. Položky se vertikálně umisťují podle účaří textu. Tabulkové bandy zpracovávají strukturovaná data, tzn. vytvoří se tolikrát, kolik je položek v dané struktuře. Více specifikací jednotlivých bandů lze opět provést pomocí jejich parametrů:

Tab. 25: Parametry tagu <band> (Vlastní zpracování)

| Název položky | Popis | | |
|---------------|--|---|-------------------------|
| Id | Označuje ID jednotlivých bandů. | | |
| Name | Jméno bandu. V případě tabulkového bandu je zde odkaz na strukturu, která se bude zpracovávat. | | |
| Height | Nastavuje výšku bandu. | | |
| Frame | Nastavuje ohraničení bandu rámečkem (je-li zadána hodnota 1). | | |
| Type | Určuje druh bandu | 0 | Hlavička |
| | | 1 | Standardní |
| | | 2 | Patička |
| | | 3 | Band - tabulka |
| | | 4 | Band – tabulka hlavička |
| | | 5 | Band – tabulka patička |
| Font | Nastavuje font použitý v bandu. | | |
| Flexible | Pokud je zadána nenulová hodnota tohoto parametru, je výška bandu počítána dynamicky z obsažených položek. | | |
| Trans | Zadává se Id transformace, která se vyvolá po začátku tisku bandu. | | |
| Page | Parametr pro hlavičkový a patičkový band, který značí číslo stránky, od které bude aktivní. | | |

| | |
|------|---|
| Down | Parametr pro patičkový band, který znamená, že patička bude přidána až na konec stránky, zbylé místo se pak doplní prázdným bandem. |
| Link | Parametr, jež zprostředkovává odkaz na data. Pokud je zadán a data existují, band je zobrazen. V opačném případě se netiskne. |

V jednotlivých bandech se pak objevují různé položky, značeny nepárovým tagem `<item />`, které mají samy o sobě velké množství parametrů a zajišťují, jak budou daná data připravena pro tisk. V následující tabulce budou uvedeny pouze základní parametry těchto položek:

Tab. 26: Parametry tagu `<item />` (Vlastní zpracování)

| Název položky | Popis | |
|---------------|--|---------------------|
| Id | Označuje ID jednotlivých položek. | |
| Name | Jméno položky. Pokud se jedná o datovou položku, určuje o která data se jedná. V případě vnořených dat se úrovně/subscripty oddělují tečkou. | |
| X | Určuje horizontální souřadnice v rámci bandu. | |
| Y | Určuje vertikální souřadnice v rámci bandu. | |
| Height | Nastavuje výšku položky. | |
| Width | Nastavuje šířku položky. | |
| Type | Určuje typ položky | 0 Textová konstanta |
| | | 1 Data |
| | | 2 Horizontální čára |
| | | 3 Vertikální čára |
| | | 4 Konstanta |
| | | 5 Obrázek |
| Value | Hodnota obsažená v položce, způsob vyhodnocení záleží na typu položky. | |
| Font | Nastavuje font použitý v položce. | |
| Link | Parametr, jež zprostředkovává odkaz na data. Pokud je zadán a data existují, band je zobrazen. V opačném případě se netiskne. | |

Zhotovená XML šablona může vypadat třeba takto:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Template>
3 <header>
4 <name>Šablona žádanky</name>
5 </header>
6
7 <const>
8 <item id="1" name="y1" value=".8" />
9 <item id="8" name="y8" value=".7" />
10 <item id="13" name="x3" value="15" />
11 </const>
12
13 <transformations>
14 <trans id="1" name="Spojení" Type="0" >
15 <titem id="1" type="0" name="Prijmeni" source="Pacient.XXX" op="1" arg1="Pacient.prijmeni" />
16 <titem id="2" type="1" name="mezera" source="Pacient.XXX" op="0" arg1=" " />
17 <titem id="3" type="1" name="Jmeno" source="Pacient.XXX" op="1" arg1="Pacient.jmeno"/>
18 </trans>
19 </transformations>
20
21 <bands>
22
23 <band id="1" type="0" height="2.1" frame="1" name="Hlavicka">
24
25 <item id="11" type="0" name="lbl_Oddeleni" x="%x1" y="%y2" height="1" width="13" value="Oddělení:" font="csr12" />
26 <item id="12" type="1" name="Zadat.zkratka" x="2.5" y="%y2" height="1" width="10" />
27 <item id="13" type="1" name="Zadat.nazev" x="%x2" y="%y2" height="1" width="13" font="csr12" />
28 <item id="14" type="0" name="lb_ICP" x="%x3" y="%y2" height="1" width="3" value="ICP:" font="csr12" />
29 <item id="15" type="1" name="Zadat.icp" x="16.1" y="%y2" height="1" width="13" />
30 </band>
31
32 </bands>
33 </Template>
```

Obrázek č. 14: Šablona pro tiskovou šablonu v XML

Takto tedy lze vytvořit tiskovou šablonu ve formátu XML, která se umístí do databáze; při požadavku na tisk žádanky je pak vybrána podle požadovaného typu, příslušný kód Caché vygeneruje podle šablony pomocí rutiny z knihovny LIBprtex, zdrojový kód pro jazyk Latex a vyvolá funkci Caché, jež z vygenerovaného zdrojového kódu vytvoří dokument ve formátu pdf, který je pak zaslán sužbou nazpět jako odpověď na kliknutí na tlačítko pro tisk žádanky.

Na následujícím obrázku je názorná ukázka vygenerovaného zdrojového kódu v jazyce Tex:

```

50 %%%%%%%%%%%
51 % Footer definition %
52 %%%%%%%%%%%
53 \def\footer2{
54 \vbox to 1.5\unit{\hrule
55 \hbox to \hsize{\vrule
56 \vbox to 1.5\unit{
57 \pointval{6}{-1.2}{1}{5}{\hfil \fxxB Patička - Strana \hfil}}
58 \pointval{12}{-1.2}{1}{3}{\the\pageno \hfil}
59 \pointhline{.5}{-.2}{15}
60 \vfil }
61 \hfil \vrule}
62 \vfil \hrule}
63 }
64 %%%%%%%%%%%
65 % Page 1 %
66 %%%%%%%%%%%
67 \header1
68 %%%%%%%%%%%
69 % Band 3 %
70 %%%%%%%%%%%
71 \vbox to 3\unit{\hrule
72 \hbox to \hsize{\vrule
73 \vbox to 3\unit{
74 \pointval{6}{-2.2}{2}{8}{%ÁDANKA O VYØETŘENÍ \hfil}
75 \pointval{0}{-2}{1}{2}{Pacient: \hfil}
76 \pointval{3}{-2}{1}{3}{SLOVÁK \hfil}
77 \pointval{6}{-2}{1}{2.4}{JAN \hfil}
78 \pointval{8}{-2}{1}{4}{RC: \hfil}
79 \pointval{10}{-2}{1}{5}{100000/0074 \hfil}
80 \pointval{10}{-4}{2}{8}{\fxxC 12345 \hfil}
81 \pointval{12}{-4}{2}{8}{12345 \hfil}
82 \vfil }
83 \hfil \vrule}
84 \vfil \hrule}

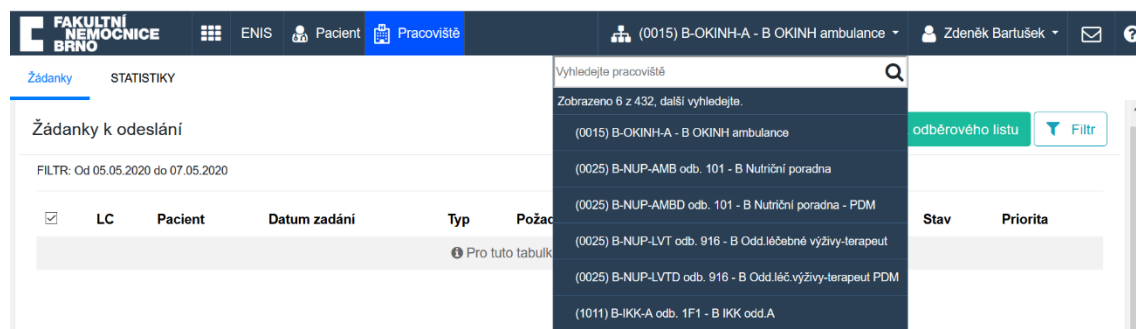
```

Obrázek č. 15: Vygenerovaný zdrojový kód v jazyce Tex (Vlastní zpracování)

3.3.1.5 Vizuální zpracování aplikace

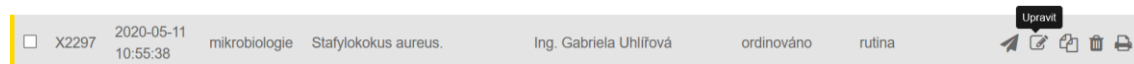
O konečnou funkčnost a podobu aplikace pro interakci s uživatelem se stará již zmiňovaný framework Angular. Po spuštění a přihlášení do aplikace se nám zobrazí záložka „Pacient“, kde zadáme požadovaného pacienta a zobrazíme se seznam jeho žádanek.

Pro zobrazení žádanky na dané pracoviště je nutné se přepnout na vedlejší kartu a v liště vybrat požadované pracoviště pro zobrazení.



Obrázek č. 19: Zobrazení z pohledu pracoviště s vyhledáním (Vlastní zpracování)

Detail žádanky pak může být zobrazen kliknutím na jednotlivé žádanky, případně mohou být provedeny další „rychlé akce“ při využití tlačítek v náhledu žádanky.



Obrázek č. 20: Zobrazení jedné žádanky ze seznamu žádank (Vlastní zpracování)

Na následujícím obrázku je pak zobrazen detail této žádanky po jejím rozkliknutí.



Obrázek č. 21: Zobrazení detailu žádanky (Vlastní zpracování)

3.4 Ekonomické zhodnocení projektu

V této části práce se budu zabývat jak finančním ohodnocením celého projektu z pohledu dodavatele, tak vzniklých nákladů pro organizaci samotnou. Zároveň zmíním ostatní benefity, které implementace tohoto informačního systému do nemocničního prostředí FN Brno přinesla.

3.4.1 Náklady řešení

Celkové ocenění tohoto systému vychází především z času stráveného na jeho vývoji a implementaci do reálného prostředí spolu s jeho podporou. V procesu vývoje jsou pak zahrnuty jednotlivé dílčí činnosti jako počáteční analýza, programování jednotlivých částí či testování.

Tab. 27: Celkové ocenění systému (Vlastní zpracování)

| | Vývoj | | | Implementace | Podpora |
|------------------------------------|-------------------|-------------------------|-----------|--------------|-----------|
| | Počáteční analýza | Proces tvorby komponent | Testování | | |
| Doba trvání (člověkohodiny) | 100 | 701 | 50 | 70 | 30 |
| Cena za hodinu | 1000 | 1200 | 800 | 800 | 700 |
| Celkem | 100 000 Kč | 841 200 Kč | 40 000 Kč | 56 000 Kč | 21 000 Kč |

Ve výše uvedené tabulce můžeme vidět přehledné ocenění celého systému rozdělené do jednotlivých fází jeho životního cyklu. Z tabulky vyplývá, že celková cena tohoto systému je 1 058 000 Kč, v ceně je pak započítáno i ocenění roční podpory tohoto systému pro případné problémy, pokud by nastaly při jeho užívání. Jiné výdaje, než je tato cena, nemocnici nevznikají, není kladen žádný další požadavek pro pořízení dalšího hardwaru pro chod tohoto systému a může být využito stávající výpočetní techniky, jež je v nemocnici implementována.

Před implementací nového systému zabraly lékařům a dalším pracovníkům nemocnice práce v souvislosti s žádankami – vytváření, editace, odesílání atp. - v průměru 5 minut na žádanku. Po implementaci a přechodu na nový systém se průměrná doba strávená nad jednou žádankou zkrátila v průměru na 3 minuty, což je téměř poloviční časová zátěž způsobovaná touto činností. Za den se v nemocnici vytvoří stovky žádanek, vezmeme-li v úvahu, že v jednom průměrném dnu se v nemocnici vytvoří zhruba 500 nových žádanek, znamená to, že před zavedením systému by doba strávená nad jejich vyřízením byla zhruba 42 hodin denně a při ocenění jedné hodiny času pracovníka v průměru na 150 Kč se jedná o částku 6 300 Kč na den. V novém systému by čas strávený touto činností

byl 25 hodin a cena by odpovídala 3 750 Kč na den, z toho plyne ušetření denních nákladů na tuto činnost v hodnotě 2 550 Kč. To by znamenalo roční úsporu 930 750 Kč, z pohledu návratnosti investice by to znamenalo, že nákup a implementace nového systému by se navrátila již během druhého roku jeho užívání, což by bylo pro nemocnici velmi pozitivní.

3.4.2 Přínosy spojené s nasazením systému

Jedním z hlavních benefitů tohoto systému je z pohledu běžných uživatelů zjednodušení manipulace a správy jednotlivých žádanek, se kterými se zde pracuje, což přináší zrychlení celého procesu. Zaměstnanci tak mohou za kratší čas vyřešit větší množství žádanek, případně mohou volný čas využít jinými činnostmi na pracovišti. Celkové hrubé vyčíslení spolu s postupem a faktory, které k němu vedly, je uvedeno v předchozí kapitole, obecně lze však odhadovat finanční úsporu na uvedených 930 750 Kč ročně.

Z pohledu nadřízených pracovníků přináší žádankový portál všechny předchozí zmíněné výhody, navíc však vzhledem k činnostem vázajícím se k jejich pracovní pozici lepší přehled nad vystavenými žádankami, kdy přesně vědí, v jakém stavu se daná žádanka nachází, ke komu se vztahuje, případně kdo a za jakým účelem ji vytvořil, vše je navíc rychle zpětně dohledatelné a transparentní.

Pro administrátory či správce, kteří se starají o udržování tohoto systému v plynulém chodu a reagují na požadavky uživatelů, se nasazením žádankového portálu zjednodušuje a zrychluje možnost těmto požadavkům vyhovět, rovněž se snižuje časová náročnost na řešení stejných druhů činností, jež musely být před nasazením žádankového portálu vykonávány delší dobu.

Ostatní výhody vzniklé implementací ZAP:

- Snížení administrativní zátěže pracovníků
- Zrychlení celého životního cyklu žádanek v nemocnici
- Zlepšení integrace systémů v organizaci
- Zvýšení bezpečnosti dat přenášených v této souvislosti

ZÁVĚR

V této práci byl kompletně navržen funkční informační systém, který umožňuje efektivnější práci s žádankami v nemocničním prostředí. V průběhu celého návrhu byl kladen důraz na splnění všech zadavatelských požadavků FN Brno, které přispěly k větší specifičnosti celé aplikace pro využívání v konkrétních podmínkách této organizace.

V první části práce je pak obsažena základní teorie potřebná pro uvedení do problematiky, případně vhodná pro lepší porozumění navrhovaného řešení, která spolu s druhou částí práce, která se zaměřuje především na analýzy spojené s FN Brno a potřebné pro „zmapování“ jejího i okolního prostředí, tvoří základ pro část třetí, jenž se zabývá návrhem konkrétního řešení vycházejícího ze zmíněných předcházejících částí.

Samotný návrh aplikace se pak skládal z několika hlavních na sebe navazujících kroků, které jsou v práci uvedeny. Nejprve bylo nutné analyzovat samotné požadavky zadavatele a z nich vytvořit základní diagram případů užití, z tohoto diagramu byly vydefinovány jednotlivé, samostatné případy užití s lepší specifikací jejich vzájemných návazností a funkcností. Od nich se pak odrážel diagram návrhu a možné realizace, který je rozdělen do dvou částí - business a aplikační, pro lepší vizualizaci celého návrhu a s ohledem na snahu o pokrytí všech procesů zmíněných v případech užití. Podle realizačního diagramu pak byly tvořeny, programovány, jednotlivé prvky této aplikace, ať už z jejího pozadí pomocí Caché či popředí pomocí frameworku Angular, které byly postupně propojovány a přidávány tak, aby odpovídaly jednotlivým případům užití a daly dohromady výsledný výstup v podobě funkční aplikace.

V mojí práci se však kromě čistého návrhu objevuje i finální ekonomické zhodnocení celého projektu a jeho využitelnost a přínosy pro FN Brno. V této závěrečné části popisují jak finanční, tak nefinanční benefity, které vznikly po nasazení této aplikace do provozu. Tyto benefity nejsou vztaženy pouze pro tuto organizaci jako celek, ale také pro jednotlivé uživatele, kteří doposud úkony tohoto charakteru zpracovávaly úplně jiným způsobem a nyní je mohou řešit touto aplikací a plně využívat všech jejích funkcionalit.

4 SEZNAM POUŽITÝCH ZDROJŮ

- [1] SKLENÁK, Vilém. Data, informace, znalosti a Internet. Vyd. 1. V Praze: C.H. Beck, 2001. ISBN 80-7179-409-0.
- [2] BRUCKNER, Tomáš. Tvorba informačních systémů: principy, metodiky, architektury. Praha: Grada, 2012, ISBN 978-80-247-4153-6.
- [3] VOŘÍŠEK, J., et al. (2008) Principy a modely řízení podnikové informatiky. 1. vyd. Praha: Oeconomia ISBN 978-80-245-1440-6
- [4] INTERSYSTEMS ONLINE DOCUMENTATION. Documentation Homepage [online] [cit. 2019-05-12] Dostupné z: <https://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls>
- [5] MÜNZ, Jan. Informační technologie ve zdravotnictví: informační systémy. V Praze: České vysoké učení technické, 2011. ISBN 978-80-01-04720-0.
- [6] LABERGE, Robert a Jakub GONER. Datové sklady: agilní metody a business intelligence. 1. vyd. Brno: Computer Press, 2012. ISBN 978-80-251-3729-1.
- [7] doplnit prezentace datové sklady
- [8] NÁRODNÍ STRATEGIE ELEKTRONICKÉHO ZDRAVOTNICTVÍ [online] [cit. 2020-01-28] dostupné z: <http://www.nsez.cz/>
- [9] FAKUTNÍ NEMOCNICE BRNO Nemocnice v datech [online] [cit. 2020-01-28] dostupné z <https://www.fnbrno.cz/nemocnice-v-datech/t3869>
- [10] MINISTERSTVO ZDRAVOTNICTVÍ ČESKÉ REPUBLIKY: Zdraví 2020: evropská zdravotní politika [online]. [cit. 2020-01-28]. Dostupné z: http://www.mzcr.cz/verejne/dokumenty/ramcove-souhrn-opatreni-zdravi2020_8526_3016_5.html
- [11] FAKUTNÍ NEMOCNICE BRNO Výroční zpráva – Fakultní nemocnice Brno [online] [cit. 2020-01-28] dostupné z <https://www.fnbrno.cz/vyrocní-zprava-2018/f4501>

5 SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

FN – Fakultní nemocnice

IS – Informační systém

ZAP – Žádankový portál

NIS – Nemocniční informační systém

ICT – Informační a komunikační technologie

BI – Business intelligence

EHR - Electronic Health Record - elektronický zdravotní záznam

6 SEZNAM POUŽITÝCH OBRÁZKŮ

| | |
|--|----|
| Obrázek č. 1: Architektura datového skladu (Vlastní zpracování) | 22 |
| Obrázek č. 2: Schéma komponent uvnitř Ensemblu (Zdroj: 4) | 25 |
| Obrázek č. 3: Diagram pro aplikaci ZAP (Vlastní zpracování)..... | 39 |
| Obrázek č. 4: Diagram zobrazující business vrstvu aplikace ZAP (Vlastní zpracování) | 40 |
| Obrázek č. 5: Diagram případů užití pro aplikaci ZAP (Vlastní zpracování) | 41 |
| Obrázek č. 6: Diagram pro aplikační vrstvu aplikace (Vlastní zpracování) | 60 |
| Obrázek č. 7: Nastavení a komponenty produkce (Vlastní zpracování)..... | 61 |
| Obrázek č. 8: Proces SendOrder (Vlastní zpracování) | 61 |
| Obrázek č. 9: Nastavení rozhraní v Management portálu (Vlastní zpracování)..... | 62 |
| Obrázek č. 10: Kód routeru pro volání jednotlivých metod (Vlastní zpracování) | 63 |
| Obrázek č. 11: Kód metody pro službu vytvoření nové žádanky (Vlastní zpracování) . | 64 |
| Obrázek č. 12: Soubor pro nastavení adres v Angularu (Vlastní zpracování)..... | 64 |
| Obrázek č. 13: Definice formuláře v Caché (Vlastní zpracování)..... | 66 |
| Obrázek č. 14: Šablona pro tiskovou šablonu v XML..... | 71 |
| Obrázek č. 15: Vygenerovaný zdrojový kód v jazyce Tex (Vlastní zpracování) | 72 |
| Obrázek č. 16: Zobrazení žádanek na pacienta v aplikaci (Vlastní zpracování) | 73 |
| Obrázek č. 17: Nabídka možných žádanek pro vytvoření (Vlastní zpracování) | 73 |
| Obrázek č. 18: Filtr pro selekci žádanek (Vlastní zpracování)..... | 73 |
| Obrázek č. 19: Zobrazení z pohledu pracoviště s vyhledáním (Vlastní zpracování)..... | 74 |
| Obrázek č. 20: Zobrazení jedné žádanky ze seznamu žádanek (Vlastní zpracování) | 74 |
| Obrázek č. 21: Zobrazení detailu žádanky (Vlastní zpracování)..... | 74 |

7 SEZNAM POUŽITÝCH TABULEK

| | |
|--|----|
| Tab. 1: Rozložení a počty obyvatel v Brně v posledních letech (Vlastní zpracování) ... | 28 |
| Tab. 2: Počet ambulantních ošetření ve FN Brno za jednotlivé roky (Vlastní zpracování) | |
| | 28 |
| Tab. 3: SWOT analýza (Vlastní zpracování) | 35 |
| Tab. 4: Příklad užití pro přihlášení k ZAP (vlastní zpracování) | 42 |
| Tab. 5: Příklad užití pro vyhledání v aplikaci ZAP (vlastní zpracování) | 43 |
| Tab. 6: Příklad užití pro editaci žádanky (vlastní zpracování) | 44 |
| Tab. 7: Příklad užití pro zobrazení žádanky (Vlastní zpracování)..... | 46 |
| Tab. 8: Příklad užití pro zobrazení žádanek s parametrem (Vlastní zpracování) | 47 |
| Tab. 9: Příklad užití pro kopii žádanky (Vlastní zpracování) | 48 |
| Tab. 10: Příklad užití pro smazání žádanky (Vlastní zpracování) | 50 |
| Tab. 11: Příklad užití pro založení nové žádanky (Vlastní zpracování) | 51 |
| Tab. 12: Příklad užití pro ověření identity (Vlastní zpracování) | 53 |
| Tab. 13: Příklad užití pro tisk žádanky (Vlastní zpracování) | 54 |
| Tab. 14: Příklad užití pro odeslání žádanky (Vlastní zpracování)..... | 55 |
| Tab. 15: Příklad užití pro tvorbu nových typů žádenek (Vlastní zpracování) | 57 |
| Tab. 16: Příklad užití pro tvorbu nových tiskových šablon (Vlastní zpracování) | 58 |
| Tab. 17: Příklad užití pro práci s aplikací ZAP z napojeného IS (Vlastní zpracování) ... | 59 |
| Tab. 18: Tabulka parametrů tagu Route (Vlastní zpracování)..... | 62 |
| Tab. 19: Parametry tagu <Form> (Vlastní zpracování)..... | 65 |
| Tab. 20: Parametry tagu <Field> (Vlastní zpracování) | 65 |
| Tab. 21: Parametry tagu <FormMap> (Vlastní zpracování)..... | 66 |
| Tab. 22: Parametry tagu <Map> (Vlastní zpracování) | 67 |
| Tab. 23: Parametry tagu <transformations> (Vlastní zpracování) | 68 |
| Tab. 24: Položky tagu <transformations> (Vlastní zpracování)..... | 68 |
| Tab. 25: Parametry tagu <band> (Vlastní zpracování)..... | 69 |
| Tab. 26: Parametry tagu <item /> (Vlastní zpracování) | 70 |
| Tab. 27: Celkové ocenění systému (Vlastní zpracování) | 75 |